AD/A-006 836

# SYNTHESIS OF STEADY-STATE SIGNAL COMPONENTS BY AN ALL-DIGITAL SYSTEM

Robert H. Davis

Naval Ordnance Laboratory
White Oak, Maryland

5 December 1974

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NOLTR 74-215

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
AD/A-006836

4. TITLE (*and Subtitle*)
Synthesis of Steady-State Signal Components by an All-Digital System

5. TYPE OF REPORT & PERIOD COVERED
Final

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Robert H. Davis

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Surface Weapons Center
White Oak, Silver Spring, MD 20910

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
A370-370K/W2144-170

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
5 December 1974

13. NUMBER OF PAGES
106

14. MONITORING AGENCY NAME & ADDRESS (*if different from Controlling Office*)

15. SECURITY CLASS. (*of this report*)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)
Signal Synthesis, Simulation, Digital, Signal Processing

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

Various algorithms for generating steady state signals by an all-digital synthesizing system are presented and analyzed. The emphasis is on evaluating the results obtained with synthesis algorithms which employ approximations or short-cuts to decrease implementation costs. Algorithms for generating both broadband and discrete or narrowband signal components using Fast Fourier Transform (FFT) techniques are investigated, and the output spectra for the approximate techniques are derived and compared to the spectra obtained with direct approaches. Finally an algorithm for the generation of arbitrary periodic signals in the time domain is presented and the spectrum of its output derived.

Preface

5 December 1974

This report is an analysis of several signal generation algorithms which were proposed and in some cases used in the Acoustic Signal Generation System (ASGS), a multi-minicomputer simulation facility developed at NAVSURFWPNCEN. In addition to deriving the spectrum of the output for each of the algorithms and comparing their performance, practical aspects of implementation for each are discussed. The report will be of interest to those concerned with the design of digital signal generation systems.

The research reported herein was partially funded under the Digital Acoustic Simulator System, Task Number A370-370K/W2144-170.

In its original form, this paper was submitted to the Graduate Faculty of the University of Maryland in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering.

ROBERT WILLIAMSON II

Michael H Stripling
MICHAEL H. STRIPLING
By direction

# TABLE OF CONTENTS

## LIST OF ILLUSTRATIONS

## LIST OF ILLUSTRATIONS (Cont.)

## GLOSSARY

| | |
|---|---|
| DFT | Discrete Fourier Transform |
| FFT | Fast Fourier Transform |
| $h(t)$ | Impulse response of a linear system |
| $H(f)$ | System function = Fourier Transform of $h(t)$ |
| IDFT | Inverse Discrete Fourier Transform |
| IFFT | Inverse Fast Fourier Transform |
| $p_x(x)$ | Probability Density Function of $x$ |
| PSD | Power Spectral Density |
| $R_w(\tau)$ | Temporal Autocorrelation Function of $w(t)$ |
| $s(t)$ | Sinusoidal signal or approximation |
| $w(t)$ | A continuous time function |
| $W(f)$ | The Fourier Transform of $w(t)$ |
| $w^x(t)$ | An impulse sampled modulation of $w(t)$ |
| $W^x(f)$ | The Fourier Transform of $w^x(t)$ |
| $w(n)$ | The value of the $n^{th}$ sample of $w^x(t)$ |
| $W(k)$ | The value of the $k^{th}$ sample of an impulsive spectral function |
| $\underset{\sim}{y}$ | Random variable |
| $\underset{\sim}{y}(t)$ | Random function defined on t |
| $Y^*$ | Complex conjugate of $Y$ |
| $\phi_{\underset{\sim}{y}}(\tau)$ | Statistical Autocorrelation Function of $\underset{\sim}{y}(t)$ |
| $\Phi_{\underset{\sim}{y}}(f)$ | Power Spectral Density of $\underset{\sim}{y}(t)$ |
| $\delta(f)$ | Impulse of unit area at $f=o$ |

# CHAPTER I

# INTRODUCTION

The synthesis of specific analog signals has been accomplished by analog systems for some time. Difficulty in control over such systems, the inherent lack of uniformity and reproducibility in their operation, and other factors such as reliability problems have, however, limited the degree of sophistication which can be achieved in analog signal synthesis. In addition, there is a lack of multiplexing capability, so that many parallel replicas of a circuit are generally required. Hybrid systems employing parametrically programmable analog devices controlled by a computer are able to surmount some of the difficulties of purely analog systems, but still have problems in reliability and reproducibility of results.

Although signal synthesis by an all digital system has been possible for a decade or more, the raw computing power necessary to perform a reasonably sophisticated job in real time has only recently become economically practical. Non real time techniques are more often than not impractical from a user's time standpoint (only in rare cases is one willing to spend days generating minutes of signal or less), or completely inapplicable for on line requirements. The recent availability of inexpensive, very fast and very reliable digital hardware, together with recent advances in digital signal processing techniques such as the FFT algorithm, have resulted in the feasibility of very sophisticated signal synthesis by all digital systems. By utilizing pseudo-random sequence generators to create the "random" components of signals, completely controllable and reproducible results may be had.

As might be expected, the increased versatility and reliability that is achievable with an all digital system is not completely free from encumbrances. Since all representations of time functions and/or of spectral functions must be discrete and of finite duration, aliasing problems are inherent in every operation. In addition, various "clever" implementation algorithms that appear on the surface to be economically desirable usually compound the sampled data system problems, and leave considerable doubt as to whether the resultant synthesized signal has the desired characteristics. It is these latter considerations to which this paper is addressed.

In general, signals may have both transient components (finite energy) and steady state components (finite power). This paper is concerned with the synthesis of steady state components, and in particular those that are either composed of one or more discrete frequency elements, or are random processes characterized by a power density spectrum which varies relatively slowly in frequency. These two categories will be referred to as narrowband and broadband components, respectively. It will be assumed that during synthesis of narrowband components, the specified frequency may be randomly perturbed by a small percentage to generate components with narrow but non-zero bandwidths.

The signals to be synthesized are divided into three groups. Chapter 2 discusses the generation of the broadband signals characterized by a finite, reasonably well behaved power spectral density. Chapter 3 considers the generation of signals having discrete or quasi-discrete spectra, and in particular the generation of sums of such signals by inverse FFT techniques. As an alternative, Chapter 4 presents a time domain technique for sums of single frequency signals whose frequencies are harmonically related; that is, all signal frequencies in a set are integer multiples of some fundamental frequency $f_o$. In all cases the objective is to develop and evaluate the performance of practical, economically implementable algorithms rather than to concentrate on determining algorithms that are optimum from a purely mathematical viewpoint.

The glossary on page vi summarizes the symbols used in the thesis. In addition, use is made of the "sinc" function, defined by

$$\text{sinc } u = \frac{\sin \pi u}{\pi u}$$

since the form (sin$\pi$u)/$\pi$u is encountered very frequently in "blocked" data (i.e., FFT) processing. A similar form also encountered quite often is (sin$\pi$u)/(sin$\frac{\pi}{N}$u), which will be designated in this paper as the "sind" function;

$$\text{sind}_N(u) = \frac{1}{N} \frac{\sin \pi u}{\sin \frac{\pi}{N} u} \qquad 1.2$$

which is shown in Appendix A to result from a finite sum of exponentials

$$\sum_{n=0}^{N-1} e^{-j2\frac{\pi n}{N}u} = N \cdot e^{-j\pi(\frac{N-1}{N})u} \quad \text{sind}_N(u) \qquad 1.3$$

Appendix B derives a useful pair of relations between the sind function and infinite sums of regularly spaced sinc functions.

It will be assumed at the outset that the functions dealt with in the text are sufficiently well-behaved to permit the interchange of summation and integration operations as required, and other minor mathematical liberties which are taken on occasion.

# CHAPTER II

# BROADBAND SIGNAL SYNTHESIS

## A. General Considerations

The problem considered in this chapter is the generation or synthesis by a digital system of random signals having a power spectral density PSD that is controllable over a band of interest W from -B to B Hertz. Since a completely arbitrary control capability would imply an infinite number of parameters to specify, it will be assumed that the desired degree of detail will be commensurate with a set of N or less control parameters $Y_k$, k = 0, ..., N-1. The objective is the determination of a signal generation algorithm defined on the $Y_k$ such that the PSD of the generated signal y(t) over the required band is a reasonably well behaved function having the desired overall form. As an example, the signal generation algorithm could be passing white Gaussian noise through a general digital filter of order N, with the $Y_k$ determining the pole/zero locations of the filter.

A further assumption is made that the desired first order statistics of the output signal be Gaussian. This is a normal requirement for most applications, since natural processes tend to exhibit Gaussian statistics.

For this reason it will also be assumed that the digital synthesizer system hardware is capable of generating samples from a Gaussian process efficiently. Actually, for most practical applications, an approximation to the Gaussian is adequate, such as the sum of 4 or more independent samples from a uniform process.

If white Gaussian noise (WGN) is all that is required for the output signal (power density spectrum equals a constant across the band of interest), then the most straightforward and efficient method of generating the desired signal is to output samples directly from the Gaussian generator, with at most a multiplication by a scale factor to adjust the output power level. Since the Fourier transform of a white Gaussian process is also a white Gaussian process (exhibiting the proper symmetries if the input is real), specification of the signal in the frequency domain (to be followed by an inverse Fourier transform) requires the same number of independent Gaussian samples as in the time domain. Thus the computational effort required to perform the inverse Fourier transform is superfluous.

If, on the other hand, a Gaussian signal with a colored spectrum is desired, both time domain techniques utilizing recursive or convolutional filters and frequency domain techniques utilizing the discrete Fourier transform have their application. If a relatively simple spectral shaping is required, passing the white Gaussian (time) samples obtained directly from the generator through a time domain filter may be adequate. This approach, however, has two rather severe limitations. The computational load exceeds the equivalent FFT implementation load if the filter function required has more than about 5 to 10 poles or zeroes. Worse yet, design of a time domain filter to implement a given frequency domain specification is not straightforward, and may become impractical if the spectral shaping must be varied in real time. The time-domain filtering approach is practical then only if very simple shaping is desired, or most profitably, if the desired shaping is naturally specified in terms of a few recursive filters (high, low or bandpass functions with rolloff in multiples of 6 db/oct.) Computational load for convolutional filters becomes excessive beyond a few zeroes, so that the required convolution is often performed by transforming to the frequency domain, multiplying, and transforming back. This of course becomes the same thing as frequency domain filtering.

If considerable detail in the spectral power density is required, frequency domain filtering techniques utilizing an FFT processor become quite attractive. The desired power spectral density can be specified at N equally spaced points from -B to B in the frequency domain, or if an impulse response or autocorrelation function is given, the equivalent spectral weighting function can be easily obtained by a forward transform. Once some threshold value of N has been passed and the decision to use FFT techniques has been made, the computational load is only weakly dependent on degree of detail implemented. The major disadvantage of the FFT methods, though, is the fact that since processing must be done in blocks rather large amounts of memory are required. The memory requirements are, in fact, proportional to the degree of detail required, i.e., the parameter N.

### B. Frequency Domain Filtering Algorithm

The most straightforward approach to synthesis of broadband signals by FFT is to implement directly the frequency filtering diagram shown in Figure 2.1. First white Gaussian noise is generated in the time domain as before. The forward transform of a block of N samples is computed, yielding N/2 single-sided complex frequency coefficients. These are then multiplied by the corresponding coefficients of the filter weighting function. The inverse FFT of the result yields the desired filtered time samples. However, to avoid erroneous results due to circular convolution, the actual filtering implementation must incorporate either of the "overlap-save" or the "overlap-add" computational techniques.[1] In either of these schemes, the desired filter impulse response of, for example, N non-zero samples must be augmented by

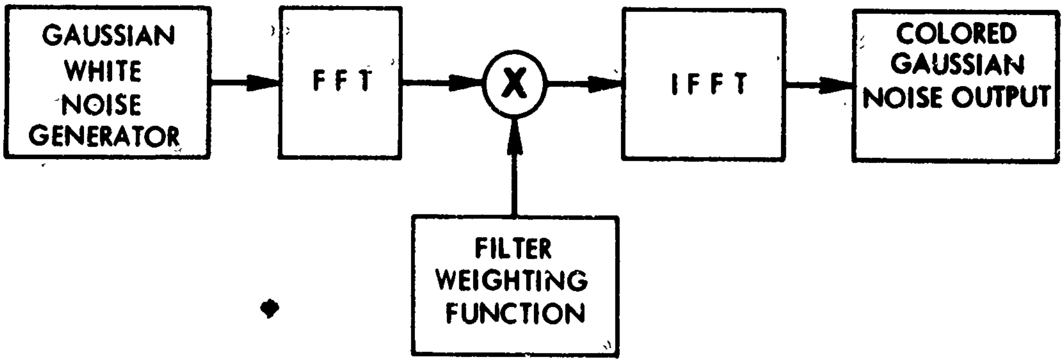
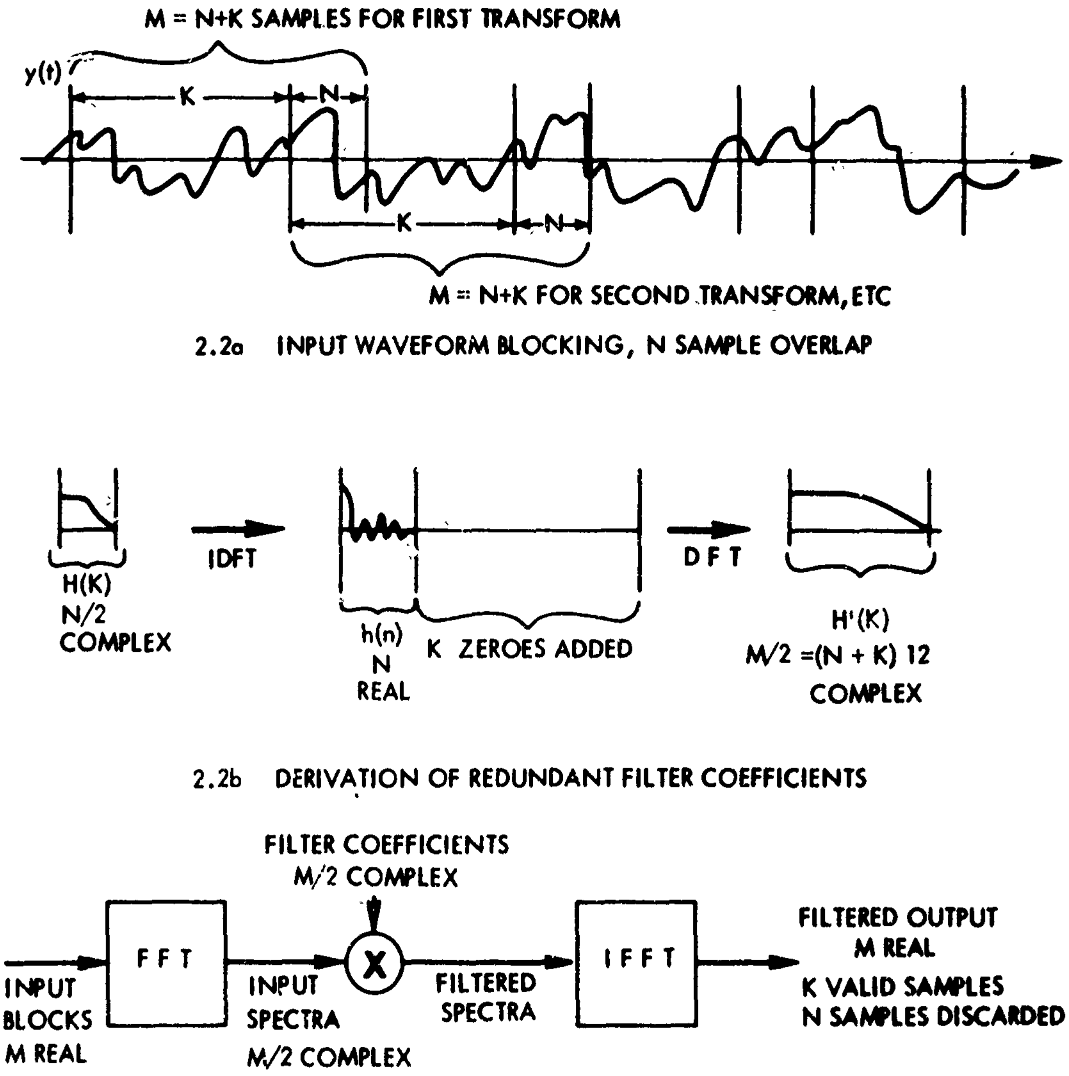

GAUSSIAN WHITE NOISE GENERATOR
FFT
X
IFFT
COLORED GAUSSIAN NOISE OUTPUT
FILTER WEIGHTING FUNCTION

FIG. 2.1 SHAPING BROADBAND SPECTRUM BY FREQUENCY DOMAIN FILTERING

a block of K zeroes to produce a total block size of M = N + K time samples. The forward transform of this function produces the spectral filter function desired having M/2 single-sided complex coefficients. Input data in blocks of M samples (or N samples plus K zeroes for the overlap-add version) are then forward transformed to produce M/2 spectral values. After multiplication of the input spectra block by the augmented filter weighting function and inverse transforming, K samples of the output are free of circular convolution error. The price to be paid, however, is increased memory size by K words, and a required overlapping of the input blocks by N samples.

The complete process is illustrated in Figure 2.2 for the overlap-save method. Note in 2.2b that to specify N parameters for the spectral detail (N/2 complex coefficients) a block size M = N + K must be used in performing the transforms. For computational efficiency, it would intuitively seem from 2.2a that K should be considerably larger than N. However, memory requirements are directly increased by increasing K, and from the memory standpoint K should be made as small as possible. In addition, the computational gains are almost non-existent for K much greater than N, (due to the log factor in the ($\frac{M}{2}\cdot\log M$) FFT load formula) and actually become losses for M more than about 8N. Figure 2.3 illustrates the computational



2.2a INPUT WAVEFORM BLOCKING, N SAMPLE OVERLAP

2.2b DERIVATION OF REDUNDANT FILTER COEFFICIENTS

2.2c OVERLAP-SAVE BLOCK DIAGRAM

FIG. 2.2 OVERLAP-SAVE FILTER ALGORITHM

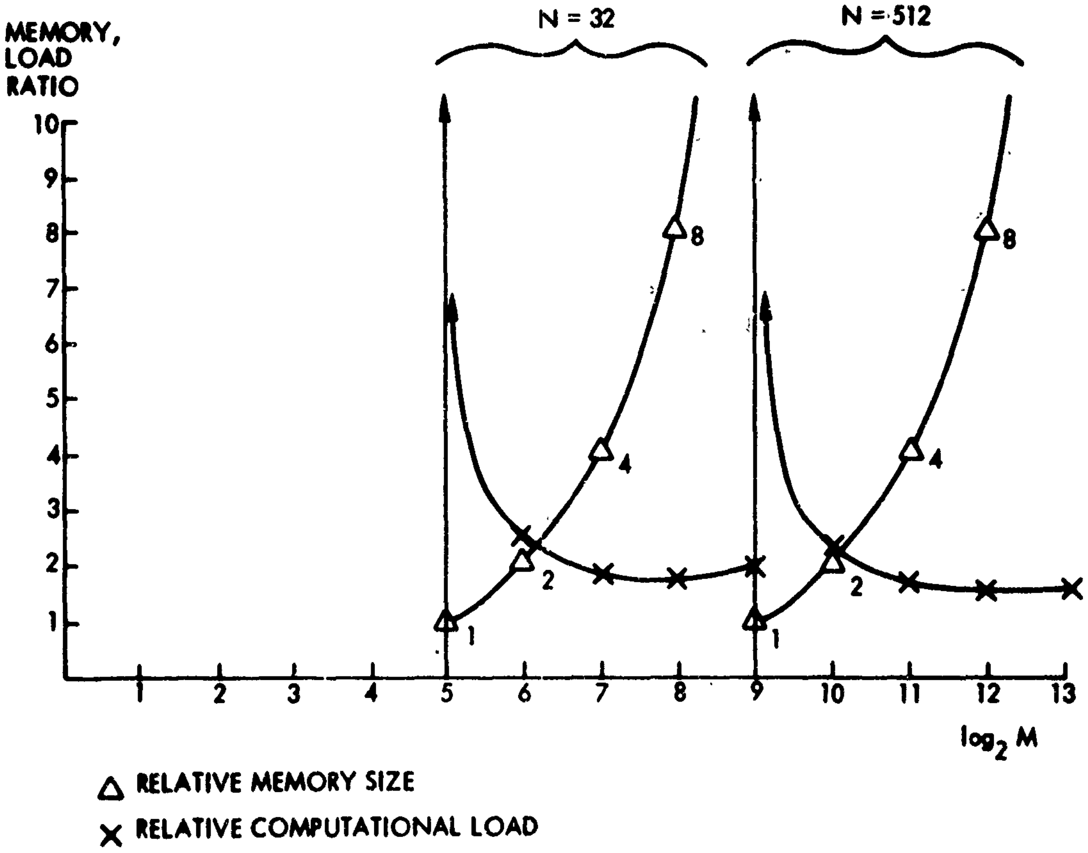


FIG. 2.3 COMPARISON OF MEMORY REQUIRED AND COMPUTATIONAL LOAD VS M FOR OVERLAP-SAVE ALGORITHM

load and memory required as a function of M for N = 32 and N = 512. For each plot, "1" is the requirement for the base transform (32, 512). In both cases M = 2N appears to be a good compromise between FFT load and memory required.

Assume that the power spectral density (PSD) $\phi(f)$ has been specified at N-1 equally spaced points covering the band of interest W, i.e.,

$$\phi(kf_c) = Y_k^{\,2}; \quad k = -\frac{N}{2} + 1, \ldots, 0, \ldots, N/2-1; \qquad 2.1$$

where N is assumed even, $Y_k = Y_{-k}$ are real and positive, and where $f_c$, the resolution, is given by

$$f_c = W/N \qquad 2.2$$

The $Y_k$ are then the magnitudes of the amplitude response at the frequencies

$$f_k = kf_c \qquad 2.3$$

The spectral response in the band of interest W at frequencies lying between the $f_k$ is determined by the particular synthesis algorithm employed to generate the output time sequence $y_n$. For the overlap-save filter algorithm described above, assume that the $Y_k$ are used directly as the N-1 non-zero spectral coefficients of the filter function, i.e.,

$$H_k = Y_k; \quad k = -N/2 + 1, \ldots, N/2 - 1 \qquad 2.4$$

where $H_{-k} = H^*_{+k} = Y_k$ since the $Y_k$ are real. The corresponding time sequence is given by

$$h_n = \sum_{k=-N/2}^{N/2-1} H_k \, e^{\frac{j2\pi nk}{N}} = \sum_{k=-N/2}^{N/2-1} Y_k \, e^{\frac{j2\pi nk}{N}}; \; n=0,\ldots,N-1 \qquad 2.5$$

where $Y_{-N/2}$ is assumed to be zero.

The filtering operation to be performed is then the convolution of a function h(t) with the input function $\underset{\sim}{x}(t)$, with

$$\underset{\sim}{x}(t) = \sum_{m=-\infty}^{\infty} \underset{\sim}{x}_m \, \delta(t - mT_s) \qquad 2.6$$

where the $\underset{\sim}{x}_m$ are the white Gaussian noise samples, and where

$$h(t) = \sum_{n=-N/2}^{N/2-1} h_n \, \delta(t - nT_s) \qquad 2.7$$

and $T_s = \frac{1}{Nf_c}$ is the basic sample period of the system.

To determine the spectrum of h(t), consider a function $h_p(t)$ which is periodic with period $NT_s = 1/f_c$ and is identically equal to h(t) over the interval $[-\frac{N}{2}T_s, \frac{N}{2}T_s]$. A function which consists of a repeating or periodic sequence of impulses has a spectrum that is also a periodic sequence of impulses, where the impulse spacing in each domain is just the reciprocal of the repeat interval in the other domain. The impulse values over one repeat interval in each of the two domains are related by the Discrete Fourier Transform (DFT) modified by a scaling factor of T. Since by definition the $h_n$ are the inverse DFT of the $H_k$, the spectrum of $h_p(t)$ is given by

$$H_p(f) = \sum_{m=-\infty}^{\infty} \sum_{k=-N/2}^{N/2-1} \frac{1}{NT_s} H_k \, \delta(f - (k+mN)f_c) \qquad 2.8$$

Thus the spectrum of $h_p(t)$ is sampled and periodic with the $H_k = Y_k$ being proportional by $1/NT_s$ to the sample values over one period $Nf_c$. But $h(t)$ is just $h_p(t)$ multiplied by a unit pulse of width $NT_s = \frac{1}{f_c}$. Therefore the spectrum of $h(t)$ is $H_p(f)$ convolved with $\frac{1}{f_c}$ sinc $f/f_c$, or

$$H(f) = \frac{1}{f_c} \sum_{m=-\infty}^{\infty} \sum_{k=-N/2}^{N/2-1} \frac{1}{NT_s} H_k \, \text{sinc}[(f - (k+mN)f_c)/f_c] \qquad 2.9$$

Interchanging sums and rewriting slightly,

$$H(f) = \frac{NT_s}{NT_s} \sum_{k=-N/2}^{N/2-1} H_k \sum_{m=-\infty}^{\infty} \text{sinc}[\frac{f}{f_c} - k - mN] \qquad 2.10$$

or, using B19,

$$H(f) = \sum_{k=-N/2}^{N/2-1} H_k \, \mathrm{sind}_N(\frac{f}{f_c} - k) \qquad 2.11$$

Since the input x(t) is assumed to be white, its PSD $\Phi_x(f)$ is just some constant which can set to unity for convenience. The PSD of the signal $y^x(t)$ at the output of the filter is then

$$\Phi_{y^x}(f) = \Phi_x(f) \cdot |H(f)|^2 \qquad 2.12$$

or

$$\Phi_{y^x}(f) = \left| \sum_{k=-N/2}^{N/2-1} H_k \; \mathrm{sind}_N(\frac{f}{f_c} - k) \right|^2 \qquad 2.13$$

Over the positive spectrum from 0 to B Hertz, $H_k$ is just equal to $Y_k$, and the spectrum of the output over the band of interest is given by 2.13. Thus the expected value of the spectral magnitude of the signal generated by the above convolution filtering algorithm passes through the values $Y_k$ at the frequencies $kf_c$, $k = -N/2$, ..., $N/2 - 1$ since $\mathrm{sind}_N(u) = 0$ for non-zero integral values of $u < N$ and exhibits sampling function interpolation at points between the $kf_c$. Note, however, that the interpolation includes contributions from the replicas of the $H_k$ outside the band of interest, and thus results in the interpolation form $\mathrm{sind}_N(u)$.

## C. Concatenated Segment Algorithm

It would seem that, since the transform of a WGN time function is a WGN spectrum (reals and imaginarys in the spectrum are all independent Gaussian random variables), it would be possible to eliminate the input forward transform and to generate the WGN spectrum directly. The problem is that, although within each block the N random variables are independent and Gaussian in both domains, there is a required correlation between blocks due to the N overlapped samples. It would be necessary to generate the spectral random variables such that the last N samples of the transform of one are identical to the first N samples of the transform of the next. It is not immediately obvious how to generate spectra with the required constraint, since the time domain is the "natural" domain for specifying the phenomena desired.

If circular convolution is completely ignored, the computational load can be reduced from one forward plus one inverse transform per output block multiplied by some factor for overlap, to just an inverse transform with no overlap. Assuming an overlap of two for the fast convolution algorithm, this corresponds to a potential reduction factor of four in computational loading. It is thus tempting to try to obtain satisfactory results by implementing the following signal generation algorithm.

Let $H_k$, $k = 0, 1, \ldots, N/2 - 1$ be the single-sided spectral coefficients of a discrete filter weighting function. Let a WGN spectrum be generated with N/2 complex single-sided frequency coefficients, $\underset{\sim}{X}_k$, where the probability density of both the real part $\underset{\sim}{X}_k^r$ and the imaginary part $X_k^i$ of $X_k$ is given by

$$p_{\underset{\sim}{X}_k^r}(X) = p_{\underset{\sim}{X}_k^i}(X) = \frac{1}{\sqrt{\pi}} e^{-X^2} \qquad 2.14$$

i.e., a zero-mean Gaussian process of variance $\sigma^2 = \frac{1}{2}$. The N random variables $\underset{\sim}{X}_k^r$, $\underset{\sim}{X}_k^i$; $k = 0, 1 \ldots N/2$, are all independent. The magnitudes of the $\underset{\sim}{X}_k$ are then Rayleigh distributed

$$p_{|\underset{\sim}{X}_k|}(r) = 2re^{-r^2};\ r \geq 0 \qquad 2.15$$

with mean $\bar{r} = \sqrt{\pi}/2$ and mean square $\overline{r^2} = 1$. The phase angles of $\underset{\sim}{X}_k$ are uniformly distributed over $2\pi$ radians:

$$p_{\underset{\sim}{X}_k}(\theta) = \frac{1}{2\pi};\ 0 \leq \theta < 2\pi \qquad 2.16$$

The signal generation algorithm consists of first forming the single-sided spectrum

$$\underset{\sim}{Y}_k = H_k \cdot \underset{\sim}{X}_k;\ k = 0, \ldots N/2 - 1. \qquad 2.17$$

Then an inverse Discrete Fourier Transform (IDFT) is performed on the $\underset{\sim}{Y}_k$ to produce the real time series $\underset{\sim}{y}_n$; $n = 0, \ldots N - 1$. The $\underset{\sim}{y}_n$ become the sample values of the output broadband process $\underset{\sim}{y}^x(t)$ for an interval $NT_s$, where $T_s$ is the basic system sample period as before. The algorithm is repeated indefinitely with a new, independent set of $\underset{\sim}{X}_k$'s on each iteration. The $\ell^{th}$ iteration generates the $\ell^{th}$ set of $\underset{\sim}{X}$'s, $\underset{\sim}{X}_{k\ell}$, from which the samples of $\underset{\sim}{y}^x(t)$ for the interval $(\ell-1/2)NT_s \leq t < (\ell+1/2)NT_s$ are obtained. Thus:

$$\underset{\sim}{y}^x(t) = \sum_{\ell=-\infty}^{\infty} \sum_{n=-N/2}^{N/2-1} \underset{\sim}{y}_{n\ell} \, \delta(t - [\ell N+n]T_s) \qquad 2.18$$

where $\underset{\sim}{y}_{n\ell}$ is the value of $\underset{\sim}{y}_n$ generated on the $\ell^{th}$ iteration.

Each segment of the output function generated in a single iteration can be thought of as exactly one period of a periodic, sampled function $\underset{\sim}{y}_{p\ell}(t)$, with sample period $T_s$ and fundamental period $NT_s$. The spectrum of such a function is also a sampled, periodic function $\underset{\sim}{Y}_{p\ell}(f)$, with period $f_s = 1/T_s$ and sample period $f_c = 1/NT_s$. Since $\underset{\sim}{y}_{p\ell}(t)$ and $\underset{\sim}{Y}_{p\ell}(f)$ are related over one period of each by the Discrete Fourier Transform, $\underset{\sim}{Y}_{p\ell}(mf_c)$ is just equal to $\underset{\sim}{Y}_{k\ell} \cdot \frac{1}{NT_s}$ for $m = k + rN$; m and r integers. Thus

$$\underset{\sim}{Y}_{p\ell}(f) = \sum_{r=-\infty}^{\infty} \sum_{k=-N/2}^{N/2-1} \frac{1}{NT_s} \underset{\sim}{Y}_{k\ell} \; \delta(f - [k + rN]f_c) \qquad 2.19$$

with $\underset{\sim}{Y}_{k\ell}$ being the $\underset{\sim}{Y}_k$ generated on the $\ell^{th}$ iteration.

The power density spectrum $\Phi_{\underset{\sim}{y}_{p\ell}}(f)$ of the function $y_{p\ell}(t)$ is just*

$$\Phi_{\underset{\sim}{y}_{p\ell}}(f) = \frac{1}{(NT_s)^2} \sum_{k=N/2}^{N/2-1} \sum_{r=-\infty}^{\infty} |\underset{\sim}{Y}_{k\ell}|^2 \, \delta(f-[k+rN]f_c) \qquad 2.20$$

The autocorrelation of the final output $\underset{\sim}{y}^x(t)$ can be found by averaging over time the statistical autocorrelation function $\phi_y(t,\tau)$. The latter is just the autocorrelation function of the member functions $\underset{\sim}{y}_{p\ell}$ if t + τ is still in the same segment as t and zero otherwise. $\phi_{y_{p\ell}}(\tau)$ is obtained by first taking the ensemble average of $\Phi_{\underset{\sim}{y}_{p\ell}}$, and then performing the Fourier transform. Applying the expectation operator to 2.20, only the $\underset{\sim}{Y}_{k\ell}$ are affected and

$$\Phi_{y_{p\ell}}(f) = \frac{1}{(NT_s)^2} \sum_{k=-N/2}^{N/2-1} \sum_{r=-\infty}^{\infty} Y_k^{\,2} \, \delta(f-[k+rN]f_c) \qquad 2.21$$

since

$$E(|\underset{\sim}{Y}_{k\ell}|^2) = Y_k^{\,2} \qquad 2.22$$

*The interested reader can readily convince himself of the validity of 2.20 by considering the Power Spectrum represented by 2.19 and noting the independence of the $Y_{k\ell}$ for all k and $\ell$.

by definition. The time average of $\phi_y(t,\tau)$ is then just $\phi_{y_{p\ell}}(\tau)$ multiplied by the percentage of all possible times t that a delay of $\tau$ still remains in the same segment. The multiplying function is thus triangular with a value of unity at the origin (average power in $y^x(t)$ is same as average power in member functions) and decreases linearly to zero at ± the segment length $NT_s$. Thus

$$\phi_y(\tau) = \begin{cases} \phi_{y_{p\ell}}(\tau)[1-\frac{\tau}{NT_s}]; & 0 \leq \tau < NT_s \\ 0 & |\tau| \geq NT_s \\ \phi_{y_{p\ell}}(\tau)[1+\frac{\tau}{NT_s}]; & -NT_s < \tau < 0 \end{cases} \qquad 2.23$$

The output power density spectrum is then the convolution of $\phi_{y_{p\ell}}(f)$ with the transform of the triangular pulse. The latter is

$$P(f) = \frac{1}{f_c^2} \text{ sinc}^2 \frac{f}{f_c} \qquad 2.24$$

Convolving with 2.21, the PSD of the output is

$$\phi_{y^x}(f) = \frac{(NT_s)^2}{(NT_s)^2} \sum_{k=-N/2}^{N/2-1} Y_k^2 \sum_{r=-\infty}^{\infty} \text{sinc}^2(\frac{f}{f_c} - k - rN) \qquad 2.25$$

or, using B24

$$\Phi_{y^x}(f) = \sum_{k=-N/2}^{N/2-1} Y_k^{\,2} \ \mathrm{sind}_N^{\,2}\left(\frac{f}{f_c} - k\right) \qquad 2.26$$

Comparing 2.26 with 2.13, the PSD differ in that the PSD of the convolution filter algorithm is of the form "square of sum", whereas the segment function algorithm yields the form "sum of squares". Since the sind functions have zeroes at all the cell frequencies except the source of each, both algorithms yield a PSD that passes through the points $|H_k|^2$ at frequencies $kf_c$. The difference exists only for those frequencies lying between the $kf_c$. The contributions to these frequencies are due to a continuous impulse response function for the convolution filter. This response function is a sum of sind functions, but each is excited by the same random process, and therefore the appearance of the "coherent sum" form. On the other hand, however, each segment output for the second approach is obtained by summing a set of independent narrowband processes, each with a spectrum of the form $\mathrm{sind}_N(x)$ and as is expected these contributions add incoherently to form the final output. However, since the stated requirement was to produce broadband noise with a PSD that is well-behaved and continuous, and that passes through the points $|H_k|^2$ at frequencies $kf_c$, both algorithms appear to qualify. Thus if the PSD is the sole criterion to be satisfied, a considerable savings in computational load can be realized by implementing the simpler algorithm.

However, all is not well with the "short-cut" approach. Each segment of the output function is actually one period of a periodic waveform, and as such is continuous from one cycle to the next, or equivalently, from the end to the beginning. The next segment generated in this manner is completely independent of the first, and the value of the function and its derivatives at the end points will have no relationship to the coresponding values at the ends of the adjacent segments. Thus the penalty for ignoring the circular convolution problem is a discontinuity in the function and its derivatives every N output samples. It is interesting to note that the first order density function of the output $y^x(t)$ (evaluated at the sample points) is independent of position, i.e.,

$$p'_y x[y(nT_s)] = p'_y x(y) \approx \frac{1}{\sqrt{2\pi}\sigma^2} e^{\frac{-y^2}{2\sigma^2}} \qquad 2.27$$

where the "approximately equal" becomes an equality if the $\underset{\sim}{Y}_{k\ell}$ reals and imaginarys are truly Gaussian. However, although the power spectrum of the generated signal is also as desired, the fact that the time function is composed of statistically independent segments is apparent in the second order density function. Although the temporal autocorrelation function of the output

$$R(\tau) = \int_{-\infty}^{\infty} \underset{\sim}{y}^{x}(t)\ \underset{\sim}{y}^{x}(t+\tau)dt \tag{2.28}$$

(the transform of the PSD) has no indication of the anomaly, the ensemble correlation function

$$\phi(t,\tau) = E\{\underset{\sim}{y}^{x}(t)\underset{\sim}{y}^{x}(t+\tau)\} \tag{2.29}$$

is not independent of t and therefore is not equal to R(τ). Thus the output process is non-ergodic. The practical implications depend on the application, but in general the simple algorithm would not likely be satisfactory if the output is to be fed into a phase or transient sensitive processing system (such as the human ear).

### D. Overlapped Segment Algorithm

For applications in which the discontinuities in the output signal are unacceptable, various compromise algorithms are possible which require overlapping output segments and therefore increased computational load, but which at least result in continuity of the function. Consider again the overlap-add version of the convolution filter algorithm (see reference (1), p. 210) with an overlap of two, i.e., M = 2N. The filter impulse response and N points of the input waveform are both augmented by K=N zeroes. The spectra of the augmented functions have N complex coefficients over the same band that the unaugmented functions have N/2; the additional N/2 coefficients representing spectral points halfway between the original points. The new points represent interpolated values based on the $\text{sind}_N(x)$ weighting intrinsic to the discrete Fourier transform. Note that this implies that each interpolated value is obtained from a weighted average of all N/2 of the original spectral coefficients. It is obvious that performing this interpolation algorithm in the frequency domain would require considerably more computation than the FFT of the original input time function. (The filter response only needs to be transformed once at setup, so it is not a problem). However, it is possible to apply other less sophisticated interpolation algorithms, which in general result in an approximation to augmenting the input with N zeroes. Thus some improvement over the concatenated segment algorithm might be had by overlapping and combining segments that are derived by a simple interpolation scheme.

The above considerations suggest an algorithm based on an overlap-add scheme, but where the segments overlapped are still independent. Let segments of length N be generated as in the previous algorithm, but then be windowed or modulated and overlapped before combining. To prevent the output signal from exhibiting modulation related to the segment period, the modulated last half of segment $\ell$ must add to the modulated first half of segment $\ell + 1$ to yield first order statistics that are independent of position. Since the random process for each segment is independent of all other segments, the overlapping halves add incoherently. Thus an appropriate modulating function is a sine or cosine half cycle pulse that has zeroes at each end of the segment and is unity at the center. The interpolation algorithm that produces a segment modulated by such a sine pulse is easily derived.

Let $\underset{\sim}{y}_{p\ell}(t)$ be the sampled, periodic segment function derivable from the $\underset{\sim}{Y}_{k\ell}$'s by the Discrete Fourier Transform. It is required that the spectrum of $\hat{\underset{\sim}{y}}_{p\ell}(t)$ be generated; where

$$\hat{\underset{\sim}{y}}_{p\ell}(t) = \underset{\sim}{y}_{p\ell}(t) \cdot \cos \frac{\pi t}{NT_s} \, , \qquad 2.30$$

and where $T_s$ is the basic sample rate as before. Multiplication of the time functions corresponds to convolution of the transforms. The Fourier Transform of cos $\frac{\pi t}{NT_s}$ is just

$$M(f) = \frac{1}{2}[\delta(f - \frac{1}{2NT_s}) + \delta(f + \frac{1}{2NT_s})] \qquad 2.31$$

i.e., impulses at $\pm 1/2NT_s$. The cell spacing is $1/NT_s$, so M(f) consists of impulses at $\pm\frac{1}{2}$ times the cell spacing. Obviously since the function $\hat{\underset{\sim}{Y}}_{p\ell}(f)$ (spectrum of $\hat{\underset{\sim}{y}}_{p\ell}(t)$) is to be non-zero for only multiples of $f_c = 1/NT_s$, direct application of 2.31 is not useful. However, by applying a little sleight-of-hand, it is possible to obtain the spectrum of a signal that is functionally equivalent to $\hat{\underset{\sim}{y}}_{p\ell}(t)$ and that has components only at $f = kf_c$ for integer k. The trick is to redefine the $H_k$'s as the spectral magnitudes at frequencies $f = (k + \frac{1}{2})f_c$. Convolution of each of the spectra $\underset{\sim}{Y}_\ell(f)$ generated from the new $H_k$'s with M(f) yields the desired spectrum $\hat{\underset{\sim}{Y}}_{p\ell}(f)$, where each component $\hat{Y}_{p\ell}(kf_c)$ is the average of the two adjacent components of $\underset{\sim}{Y}_\ell(f)$, $\underset{\sim}{Y}_{k\ell}$ and $\underset{\sim}{Y}_{(k+1)\ell}$. The transform of $\hat{\underset{\sim}{Y}}_{p\ell}(f)$ then produces the function $\hat{\underset{\sim}{y}}_{p\ell}(t)$, of which one $NT_s$ - second segment between zeroes of the sinusoidal envelope is added to the output $\underset{\sim}{y}^{x}(t)$ being generated. The fact that the phases of the components of $\hat{\underset{\sim}{Y}}_{p\ell}(f)$ are uniformly distributed over $2\pi$ radians permits taking the $NT_s$-second segment of $\underset{\sim}{y}_{p\ell}(t)$ from $\frac{-NT_s}{2}$ to $\frac{NT_s}{2}$ with no loss in generality. To find the PSD of the output generated by the latter method, let $\underset{\sim}{Y}_{p\ell}(f)$ be written

$$\underset{\sim}{Y}_{p\ell}(f) = \sum_{r=-\infty}^{\infty} \sum_{k=N/2}^{N/2-1} \frac{1}{NT_s} \underset{\sim}{Y}_{k\ell} \quad \delta(f-(rN+k+\tfrac{1}{2})f_c ) \qquad 2.32$$

where the $\underset{\sim}{Y}_{k\ell}$ are generated from the redefined $H_k$'s as in 2.17. Convolving with M(f)

$$\hat{\underset{\sim}{Y}}_{p\ell}(f) = \frac{1}{2NT_s} \sum_{r=-\infty}^{\infty} \sum_{k=-N/2}^{N/2-1} \underset{\sim}{Y}_{k\ell}\{\delta(f-(rN+k)f_c + \delta(f-(rN+k+1)f_c\}$$

2.33

Isolating a segment of length $NT_s$ results again in convolving with $\frac{1}{f_c}$ sinc $f/f_c$, yielding

$$\hat{\underset{\sim}{Y}}_{\ell}(f) = \frac{1}{2} \sum_{r=-\infty}^{\infty} \sum_{k=-N/2}^{N/2-1} \underset{\sim}{Y}_{k\ell}\{\text{sinc}(f/f_c-rN-k)+\text{sinc}(f/f_c-rN-k-1)\}$$

2.34

or

$$\hat{\underset{\sim}{Y}}_{\ell}(f) = \frac{1}{2} \sum_{k=-N/2}^{N/2-1} \underset{\sim}{Y}_{k\ell} \sum_{r=-\infty}^{\infty} \{\text{sinc}(f/f_c-rN-k) + \text{sinc}(f/f_c-rN-k-1)\}$$

2.35

The Energy Spectral Density of $\underline{\hat{Y}}_\ell(f)$ is then

$$\Phi_{\hat{y}_\ell}(f) = E[\underline{\hat{Y}}_\ell(f)\,\hat{Y}^*_\ell(f)] \qquad 2.36$$

$$= \frac{1}{2}\sum_{k=-N/2}^{N/2-1}\sum_{j=-N/2}^{N/2-1} E[\underline{Y}_{k\ell}Y^*_{j\ell}]\cdot\sum_{r=-\infty}^{\infty}[\mathrm{sinc}(\frac{f}{f_c}-rN-k)$$

$$+\,\mathrm{sinc}(\frac{f}{f_c}-rN-k-1)]\cdot\sum_{m=-\infty}^{\infty}[\mathrm{sinc}(\frac{f}{f_c}-mN-j)+\mathrm{sinc}(\frac{f}{f_c}-mN-j-1)]$$

$$2.37$$

Again, $\underline{Y}_{k\ell}$ is independent of $\underline{Y}_{j\ell}$ unless k=j, and then $E[\underline{Y}_{k\ell}\,Y^*_{k\ell}] = H_k^2$

$$\Phi_{\hat{y}_\ell}(f)=\frac{1}{2}\sum_{k=-N/2}^{N/2-1} H_k^2\,(\sum_{r=-\infty}^{\infty}\{\mathrm{sinc}(\frac{f}{f_c}-rN-k)+\mathrm{sinc}(\frac{f}{f_c}-rN-k-1)\})^2$$

$$2.38$$

Finally, the PSD of $\underset{\sim}{y}^x(t)$ is again just twice the average over $\ell$ of 2.38 (factor of two due to overlap), or

$$\Phi_{y^x}(f) = 2\cdot\Phi_{\hat{y}_\ell}(f) \qquad 2.39$$

The form of 2.38 is the same as 2.10 and 2.25 except that the interpolating function is the sum of two sinc(x) functions displaced by unit separation. Writing the sum out,

$$\operatorname{sinc} x + \operatorname{sinc}(x + 1) = \frac{\sin \pi x}{\pi x} + \frac{\sin(\pi x + \pi)}{\pi x + \pi}$$

$$= \frac{\sin \pi x}{\pi x}\left(\frac{1}{x + 1}\right) \qquad 2.40$$

Note that the result has the same oscillatory behavior as sinc x, but falls off as $x^2$ rather than x. This is especially beneficial when it is desired to produce an output $\underset{\sim}{y}^{x}(t)$ whose spectrum has abrupt changes in level as a function of frequency.

Summarizing, the straightforward application of standard convolution filter techniques to the synthesis of a random process with specified PSD requires on the order of a factor of four more computational load than a minimum implementation ignoring circular convolution. Although the PSD at the specified points and the first order time statistics are the same, the minimum implementation output is discontinuous and may be unsatisfactory for transient-sensitive systems. A compromise solution requiring a factor of two overlap and therefore a factor of two more computational load is somewhat more satisfactory in that the output is continuous, but the derivatives at the segment ends are still discontinuous. However, it will be shown in Chapter 3 that the factor of two overlap is very desirable when discrete or line components are to be added to the generated signal.

# CHAPTER III

# IFFT GENERATION OF DISCRETE FREQUENCIES

## A. The Exact Synthesis Algorithm

In an analog system, the generation of a sinusoidal waveform is usually implemented by constructing an oscillator having the desired frequency . The direct adaptation of this approach to discrete or line frequency generation by a digital system requires a second order difference equation with a pole at the desired frequency. As a general digital synthesis technique, this approach has problems with stability due to a coefficient quantization, as well as the disadvantage that each frequency requires a separate generator. On the other hand samples of the desired sinusoid are generated one at a time, and thus for a small number of discrete frequencies desired, the meager storage requirements are attractive relative to block processing techniques. However, for the case where the sum of many discretes is desired, significant computational savings can be obtained by utilizing block processing with the Fast Fourier Transform (FFT).

Implementation of a general second-order difference equation requires about the same amount of computation as an FFT butterfly. Generation of N samples of a single sinusoid by difference equation requires N such computations, whereas generation of N samples by IFFT requires $\frac{N}{2}\log_2 N$ computations for the transform plus whatever computation is required to specify the desired sinusoid in the DFT domain. Thus, for N on the order of 1000, generation of the sum of more than 5 discretes begins to favor the FFT approach unless an unreasonable amount of computation is required to specify the lines in the sampled frequency space input to the IFFT.

For generation of discretes with frequencies that are exactly equal to the cell frequencies $f_c = f_s/N$, where $f_s$ is the sampling frequency, the specification in the sampled frequency domain is trivial. Consider the IFFT of N/2 single-sided complex frequency coefficients $F_k$, k=0,1,...N/2-1, to produce N time samples $f_n$, n=0,...N-1. A non-zero $F_k$, say $F_{k_1}$, will result in a contribution to the output time segment of exactly $k_1$ cycles of a sinusoid (period = $N/k_1$ sample periods), with a magnitude and a phase relative to the beginning of the segment determined by the complex coefficient $F_{k_1}$. Since successive segments generated with the same $F_{k_1}$ will match perfectly at the segment boundaries, the result will be a perfect sinusoid (sampled, of course) with frequency $\frac{k_1}{N}f_s$. Thus for independent discretes with frequencies in multiples of $f_s/N$, a single complex number added to the appropriate spectral cell before each inverse transform will result in the desired sinusoid added to the output time waveform.

For frequencies that are not exact multiples of $f_s/N$, the situation is considerably less straightforward. Since the discrete Fourier Transform (DFT) is information lossless and reversible, the easiest way to see what is required to generate arbitrary frequency sinusoids is to consider the forward transform of successive segments of the desired waveform. First consider the (complex to complex) DFT of an arbitrary complex exponential w(t) given by:

$$w(t) = e^{j(2\pi f_a t+\alpha)} \qquad 3.1$$

where $f_a$ is between 0 and $f_s/2$, and $\alpha$ is the phase at the beginning of the segment of length $NT_s$ from which N samples have been taken. Let $f_a$ be expressed as

$$f_a = (m + d)\frac{f_s}{N} \qquad 3.2$$

where $\frac{f_s}{N}$ is the cell spacing, and m is an integer and d less then unity. $m\frac{f_s}{N}$ is then the next lowest cell frequency, and d is the fraction of a cell spacing that $f_a$ is above $m\frac{f_s}{N}$. The sampled version of w(t) is then

$$w^x(t) = \sum_{n=-\infty}^{\infty} e^{j\alpha} e^{j2\pi(m+d)\frac{f_s}{N} nT_s} \delta(t-nT_s) \qquad 3.3$$

and the samples of the segment of interest are

$$w(n) = e^{j(\frac{2\pi}{N}(m+d)n+\alpha)}, \quad n=0,\ 1 \ \dots \ N-1 \tag{3.4}$$

where the fact that $f_s T_s = 1$ has been noted. The DFT is then

$$W(k) = \frac{1}{N} \sum_{n=0}^{N-1} e^{j\alpha} \, e^{j\frac{2\pi n}{N}(m+d)} \, e^{-j\frac{2\pi nk}{N}} \tag{3.5}$$

Removing the phase angle from the sum and combining the exponential arguments

$$W(k) = \frac{e^{j\alpha}}{N} \sum_{n=0}^{N-1} e^{-j\frac{2\pi n}{N}(k-m-d)} \tag{3.6}$$

From Appendix A,

$$\sum_{n=0}^{N-1} e^{-j2\frac{\pi}{N}nu} = Ne^{-j\pi(\frac{N-1}{N})u} \, \mathrm{sind}_N(u) \tag{3.7}$$

and

$$W(k) = e^{j\alpha} e^{-j\frac{\pi}{N}(N-1)(k-m-d)} \, \mathrm{sind}_N(k-m-d) \tag{3.8}$$

If d=0, only $W(m) = e^{j\alpha}$ is non-zero as was noted above. However, for any non-zero d, $0 < d < 1$, all W(k) are non-zero. The magnitude of the W(k) are governed by the $\text{sind}_N(k-m-d)$ form, and the phases are governed by $e^{+j(\alpha - \pi\frac{(N-1)}{N}(k-m-d))}$, where $\alpha$ is the desired phase angle at the beginning of the segment.

For a real time function

$$w(t) = e^{j(2\pi f_a t + \alpha)} + e^{-j(2\pi f_a t + \alpha)} . \qquad 3.9$$

By linearity of the DFT, the corresponding sampled spectra of a sampled segment is given by the sum of two terms at $\pm f_a$

$$W(k) = \frac{e^{j\alpha}}{N} e^{-j\frac{\pi}{N}(k-m-d)} \quad \text{sind}_N(k-m-d)$$

$$+ \frac{e^{-j\alpha}}{N} e^{j\frac{\pi}{N}(k+m+d)} \quad \text{sind}_N(k+m+d) . \qquad 3.10$$

To avoid unnecessary complication in the math, the following arguments will be in terms of generating a single complex exponential. The corresponding real sinusoid is easily obtained by summing a pair of conjugately symmetric exponentials at $\pm f_a$, the desired frequency.

The required synthesis algorithm is thus to add contributions to every spectral cell (for each discrete frequency desired) according to 3.8 (or 3.10). Since there will be exactly m+d cycles of the required sinusoid in each N sample segment, the phase angle

$\alpha$ must be increased by $2\pi d$ radians for the generation of successive segments to maintain continuity from one segment to the next.

### B. An M Cell Approximation

The computation of the W(k) for N/2 spectral cells to generate N output samples obviously requires more computational effort than direct generation of the sinusoid in the time domain, even before considering the overhead of computing the inverse DFT. Thus the exact solution using FFT techniques is not practical unless the required frequencies can be constrained to the set $f_a = nf_c$, $n=0, \pm 1, \ldots, \pm N/2-1$.

If, on the other hand, something less than perfection is acceptable, an approximate solution may yield the hoped for computational savings. Since most of the power in the spectral domain is contained in cells near the desired frequency, it would seem reasonable to approximate the complete spectral description by truncating the tails of the $\text{sind}_N(x)$ envelope or by some other equivalent operation. This could result in an acceptable number of coefficients to be specified for each segment of each discrete required in the output. If a 1024 complex to 2048 real IFFT is being utilized and if only ten non-zero coefficients per discrete will produce acceptable output results, the generation of 10 complex coefficients in the spectral domain becomes quite attractive relative to having to generate 2048 numbers in the time domain for each sinusoid desired.

Assume that only the M largest (in magnitude) contributions per complex exponential are to be retained unmodified in the sampled frequency domain. Since the contributions of all other cells are to be eliminated, the total power is decreased by the sum of the squared magnitudes of the eliminated spectral components. In fact, due to the orthogonality of the Fourier frequencies, the M largest components form a least mean squared error estimate of the original signal if only M cells may be non-zero. Thus if mean squared error is an acceptable criteria, adding the M largest contributions based on either 3.8 (for single sided spectra) or 3.10 (for double sided spectra) yields the optimum approximation.

Since the $\text{sind}_N(x)$ envelope of the spectral magnitudes decreases monotonically on either side of the desired frequency, zeroing all but the M largest components is equivalent to multiplying the spectrum by a pulse or "window" of unit magnitude and width $M\frac{f_s}{N}$. Multiplication in the frequency domain is equivalent to convolving the time function (sinusoidal segment) with the transform of the window.

The resulting time function segment can be determined by an inverse DFT of the windowed spectrum

$$s(n) = \sum_{k=m-\frac{M}{2}+1}^{m+\frac{M}{2}} W(k)\, e^{j\frac{2\pi nk}{N}} \quad . \qquad 3.11$$

Using 3.6,

$$s(n) = \sum_{k=m-\frac{M}{2}+1}^{m+\frac{M}{2}} e^{j\alpha} \sum_{\ell=0}^{N-1} \frac{1}{N}\, e^{-j\frac{2\pi\ell}{N}(k-m-d)}\, e^{j\frac{2\pi nk}{N}}$$

3.12

Interchanging the order of summation and refactoring the exponentials,

$$s(n) = e^{j\alpha} \sum_{\ell=0}^{N-1} \frac{1}{N}\, e^{j\frac{2\pi\ell}{N}(m+d)} \sum_{k=m-\frac{M}{2}+1}^{m+\frac{M}{2}} e^{j\frac{2\pi k}{N}(n-\ell)}$$

3.13

Factoring an $e^{j\frac{2\pi m}{N}(n-\ell)}$ from the last summation and modifying the summation limits correspondingly,

$$s(n) = e^{j\alpha} \sum_{\ell=0}^{N-1} \frac{1}{N}\, e^{j\frac{2\pi\ell}{N}(m+d)}\, e^{j\frac{2\pi m}{N}(n-\ell)} \sum_{k=-\frac{M}{2}+1}^{\frac{M}{2}} e^{j\frac{2\pi k}{N}(n-\ell)}$$

3.14

Factoring the original sinusoid from the summations, and reordering again,

$$s(n) = e^{j\alpha} \, e^{j\frac{2\pi n}{N}(m+d)} \sum_{k=-\frac{M}{2}+1}^{\frac{M}{2}} e^{j\frac{2\pi n}{N}(k-d)} \frac{1}{N} \sum_{\ell=0}^{N-1} e^{-j\frac{2\pi}{N}\ell(k-d)} . \qquad 3.15$$

The last sum is just the $\text{sind}_N(k-d)$, so

$$s(n) = e^{j\alpha} \, e^{j\frac{2\pi n}{N}(m+d)} \sum_{k=-\frac{M}{2}+1}^{\frac{M}{2}} e^{j\frac{2\pi n}{N}(k-d)} \, e^{-j\frac{\pi}{N}(N-1)(k-d)} \cdot \text{sind}_N(k-d). \qquad 3.16$$

or, finally, the desired form,

$$s(n) = e^{j\alpha} \, e^{j\frac{2\pi n}{N}(m+d)} \sum_{k=-\frac{M}{2}+1}^{\frac{M}{2}} e^{j\frac{\pi}{N}(k-d)(2n-N+1)} \text{sind}_N(k-d). \qquad 3.17$$

This is just the desired sinusoidal segment, but modulated by a gain factor which is dependent on both n and d.

An example is helpful in illustrating the result of the uniform window approximation. Figure 3.1 shows a 128 sample function segment which covers 10.25 cycles of a sinusoid. Figure 3.2 is the corresponding 64 point single sided spectra derived by DFT. Note the 180°

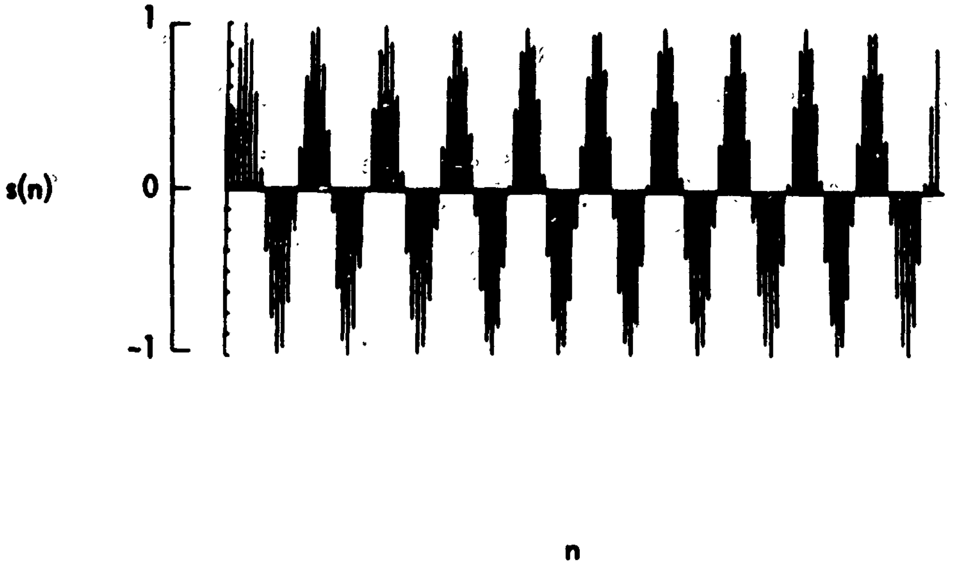


FIG. 3.1 SAMPLED SINUSOID SEGMENT - 10.25 CYCLES, 128 SAMPLES

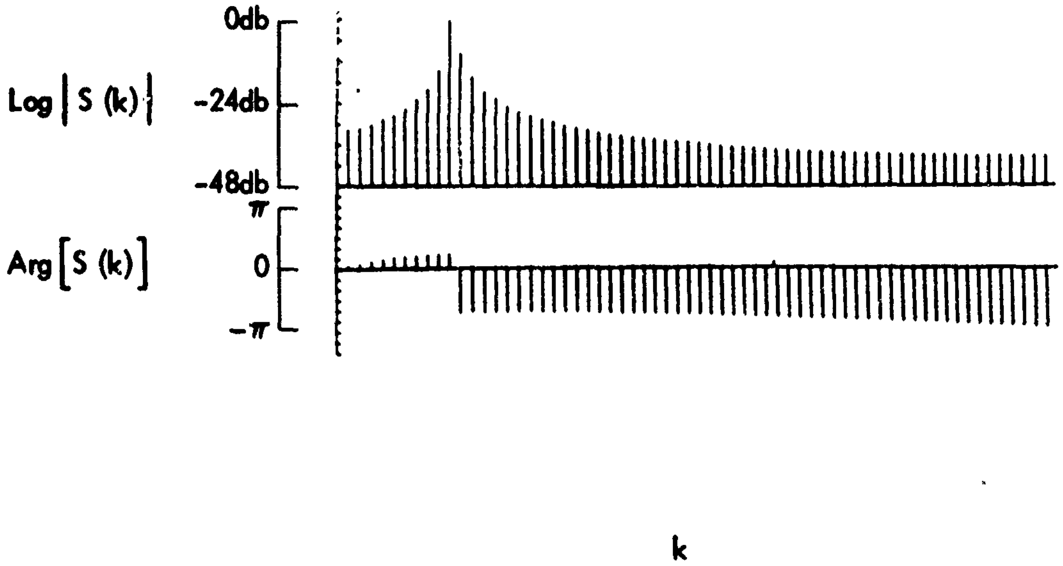


FIG. 3.2 LOG MAGNITUDE-PHASE SPECTRUM OF 10.25 CYCLE SEGMENT.

phase difference for components just above and just below the actual frequency. A uniform window of width M=10 is applied to produce the approximation spectra in Figure 3.3. An inverse DFT then gives the approximation time segment shown in Figure 3.4.

Crudely speaking, the major effect of truncating the tails of the segment spectrum is to attempt to "match up" the two ends of the segment. This is to be expected since the "end-around" discontinuity, or equivalently, the discontinuity from one "cycle" to the next of the periodic waveform represented, cannot be sustained without the frequency components contained in the tails of the spectrum. The error for the example given is shown in Figure 3.5, and is representative of the relative size of the error at the ends of a segment versus the center portion.

A composite time segment was constructed by concatenating four segment approximations derived as in the example above, but with initial phases of 0, $\pi/2$, $\pi$, and $3\pi/2$ radians, respectively. The resultant time block contains 512 samples covering a total of 41 cycles of the desired sinusoid, and is shown in Figure 3.6. Note the rather severe effective modulation of the signal envelope in the vicinities of the segment boundaries. The error function for the composite waveform is shown in Figure 3.7, and has a maximum value of .55 relative to a perfect (all

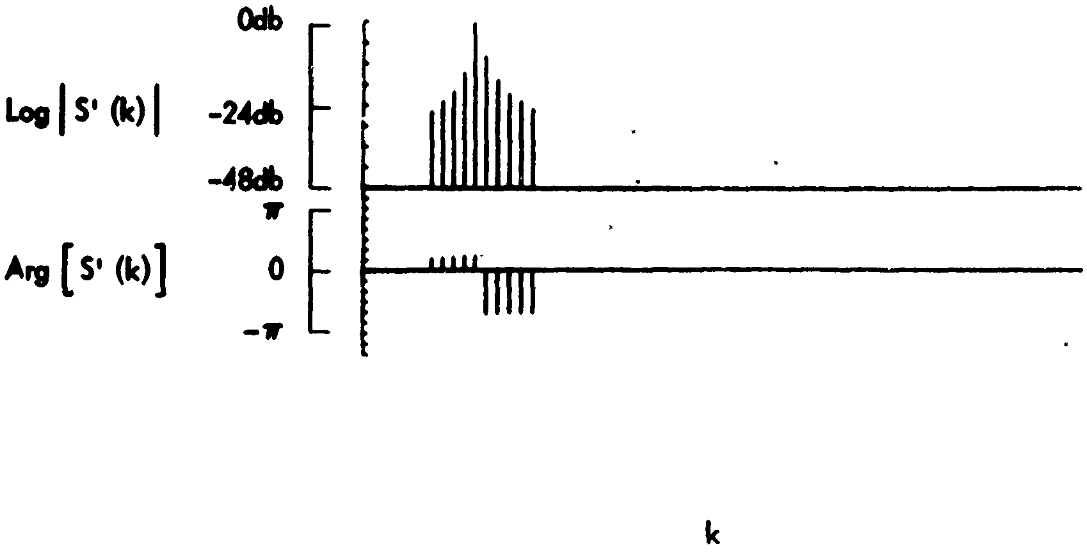


FIG. 3.3 TRUNCATED SPECTRUM OF APPROXIMATION SEGMENT

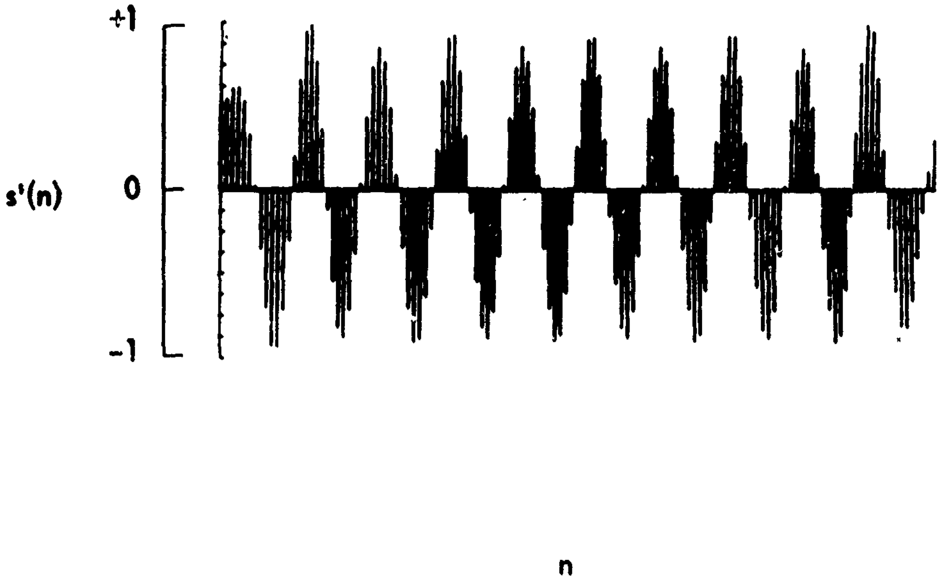


FIG. 3.4 APPROXIMATION SEGMENT WAVEFORM

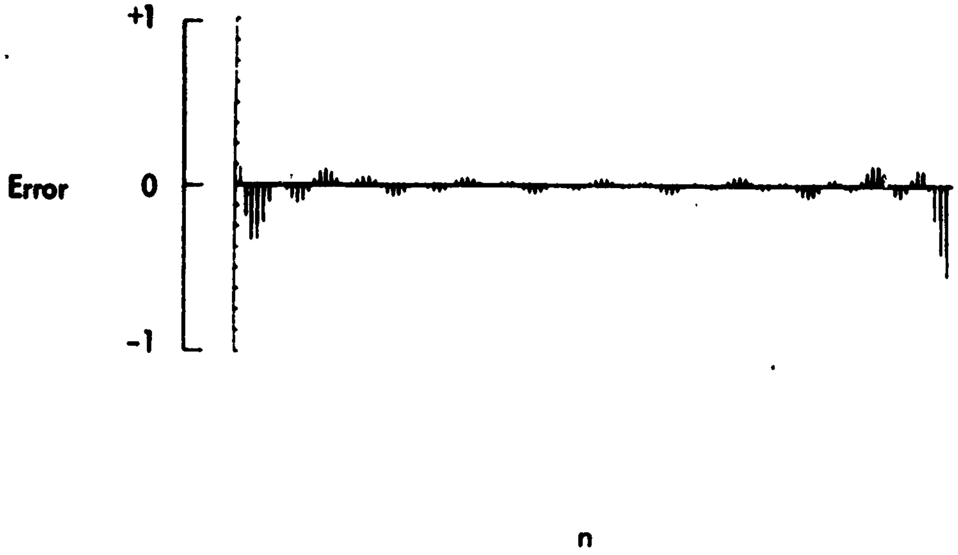


FIG. 3.5 SEGMENT ERROR s'(n) - s(n).

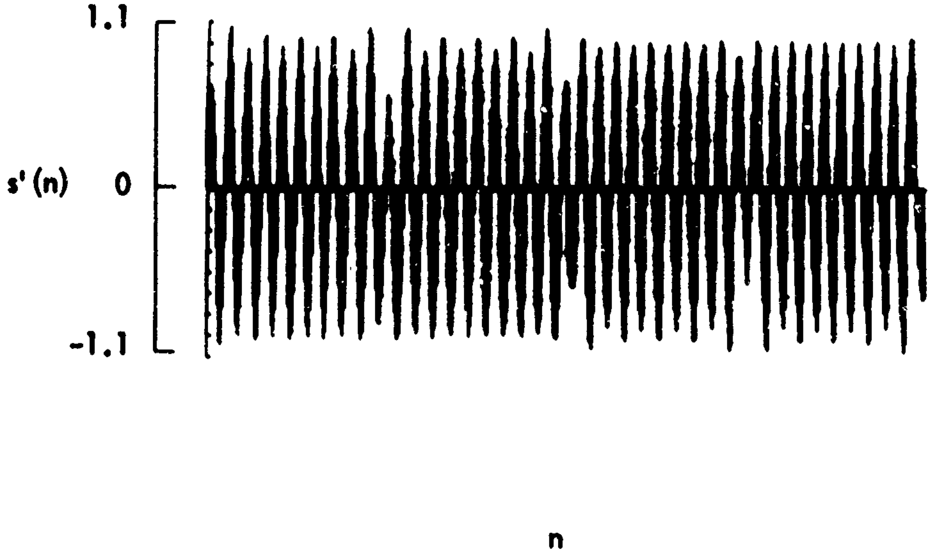


FIG. 3.6 COMPOSITE 512 SAMPLE APPROXIMATION WAVEFORM.

spectral components preserved) waveform amplitude of ±1. The maximum excursion of the approximation waveform has a relative amplitude of 1.09.

The spectrum of the composite approximation waveform is shown in Figure 3.8. The largest undesired component is about 24 db below the desired signal strength. Note that the major error components are at frequencies $f_a \pm n/NT_s$, where $f_a$ is the desired frequency and $NT_s$ is the length of each approximation segment. This is the behavior expected in the frequency domain corresponding to the effect described above of the time waveform being effectively modulated on a once per segment rate.

If the criterion of acceptability is other than minimum mean square error, such as ratio of desired signal to largest undesired component or minimum amplitude modulation of the time function, the uniform window may not be the best solution. Shading of the window coefficients was briefly pursued, but in general decreased side lobe strengths were obtainable only at the expense of increases in amplitude modulation in the time function. In addition, the fact that the spectral truncation operation reduces the "end-around" discontinuity for each segment means that discontinuities are created in the composite output waveform at the segment boundaries. Although increased fidelity can be achieved by using a wider window, any computational advantages of the technique are soon lost.

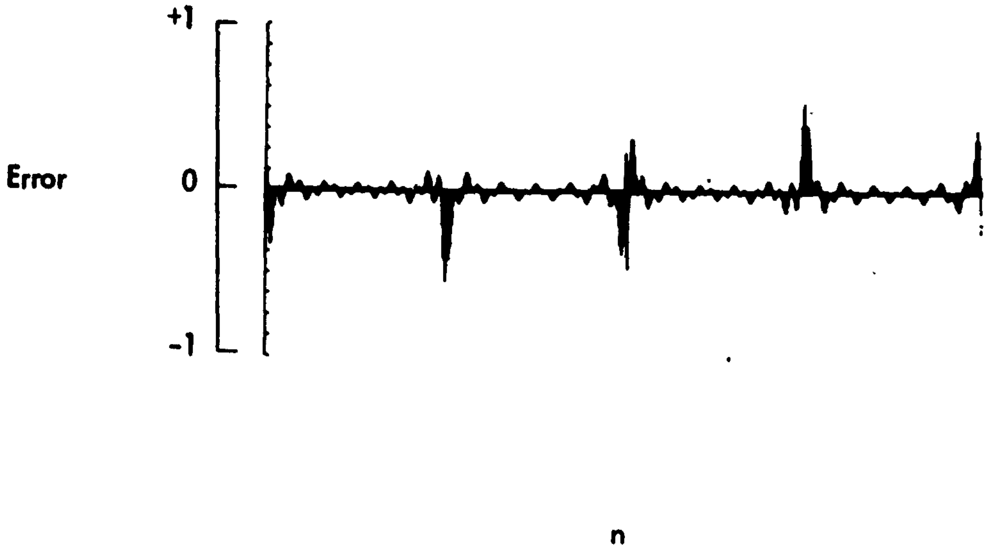


FIG. 3.7 COMPOSITE APPROXIMATION ERROR.

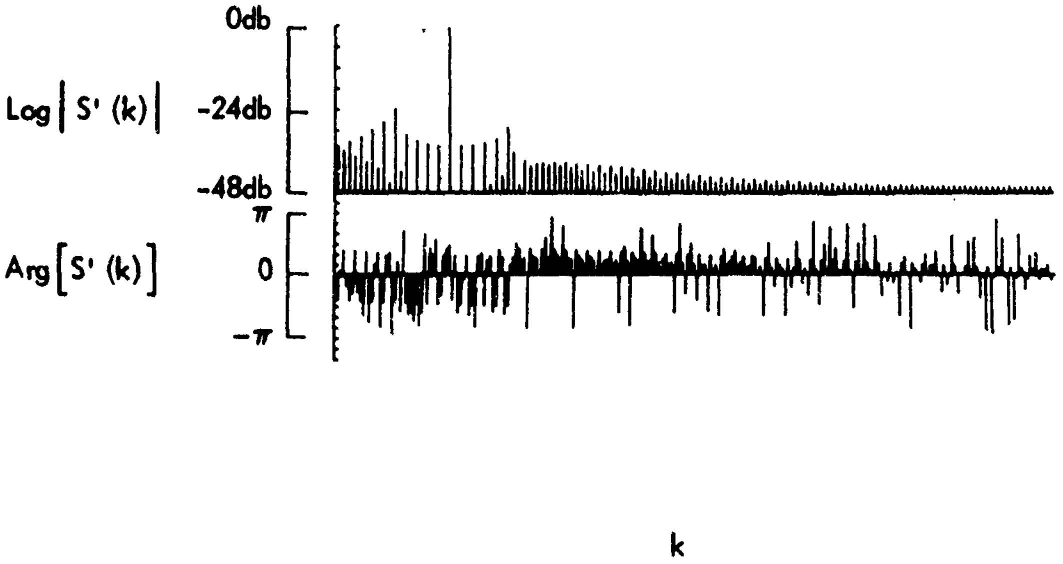


FIG. 3.8 COMPOSITE APPROXIMATION SPECTRUM.

It appears, then, that this particular FFT based algorithm is not very useful unless a rather coarse approximation is acceptable, preferably with only a mean square error fidelity requirement.

### C. The "Overlapped-Hanned" Algorithm

The fact that the major difficulty encountered in the above algorithm was an effective modulation of each segment suggests the possibility of purposely introducing a specific modulation characteristic and to compensate by overlapping and summing adjacent segments; i.e., introduce redundant processing. This is all the more attractive if the desired discrete components are to be combined with a specified broadband signal, since it was shown in Chapter 2 that a two-times redundancy factor yielded a fairly satisfactory broadband algorithm. If the same factor of 2 can be exploited for the discrete case, the approximation coefficients can be added to the broadband spectra before transforming, and thus one IFFT does all.

Since the contributions in each of the overlapped segments add coherently to form the total signal for the discrete case, the sine pulse weighting used for the broadband algorithm is not applicable. A triangular pulse modulation form meets the requirement in the time domain, but is not obtainable by simple manipulation of the spectrum of the signal. On the other hand, the raised cosine modulation (corresponding to "Hanning" shading) of the segment appears to be an almost perfect solution to the problem. Overlapped signal segments that have been modulated by the Hanning pulse $p_H(t)$

$$p_H(t) = \begin{cases} .5 + .5 \cos 2\pi t/NT_s; \ |t| < NT_s/2 \\ 0; |t| \geq NT_s/2 \end{cases} \qquad 3.18$$

will add perfectly to produce the original waveform. The question thus becomes: can the Hanned segment be more easily approximated than the unmodulated segments?

Returning to the spectrum of the unweighted segment, Equation 3.8, the spectrum of the Hanned segment $W_H(k)$ is obtained by convolving with the transform of $\frac{1}{2}(1 + \cos 2\pi\ n/N)$. The latter is the well known $P_H(k)$,

$$P_H(k) = \frac{1}{4}\ \delta(k+1) + \frac{1}{2}\ \delta(k) + \frac{1}{4}\ \delta(k-1) \qquad 3.19$$

i.e., impulses at zero frequency and at ± one frequency cell $\frac{f_s}{N}$.

Convolving 3.8 with 3.19 the result is

$$W_H(k) = e^{j\alpha} \; e^{j\pi(\frac{N-1}{N})d}\Bigg[(\tfrac{1}{2}\, e^{-j\pi(\frac{N-1}{N})(k-m)}\,\mathrm{sind}_N(k-m-d)$$

$$+ \tfrac{1}{4}\, e^{-j\pi(\frac{N-1}{N})(k-m+1)}\,\mathrm{sind}_N(k+1-m-d) + \tfrac{1}{4}\, e^{-j\pi(\frac{N-1}{N})(k-m-1)}$$

$$\mathrm{sind}_N(k-1-m-d)\Bigg] \qquad 3.20$$

Manipulating and writing out the sind functions,

$$W_H(k) = \frac{1}{2N}\, e^{j\alpha} \; e^{j\pi(\frac{N-1}{N})d} \; e^{-j\pi(\frac{N-1}{N})(k-m)}\Bigg[\frac{\sin\pi(k-m-d)}{\sin\frac{\pi}{N}(k-m-d)}$$

$$+ \frac{1}{2}\, e^{-j\pi(\frac{N-1}{N})} \; \frac{\sin[\pi(k-m-d)+\pi]}{\sin[\frac{\pi}{N}(k-m-d+1)]} + \frac{1}{2} \; e^{j\pi(\frac{N-1}{N})}$$

$$\frac{\sin[\pi(k-m-d)-\pi]}{\sin[\frac{\pi}{N}(k-m-d-1)]}\Bigg] \qquad 3.21$$

In the last two terms, $\sin[\pi(k-m-d)\pm\pi]$ is just $-\sin[\pi(k-m-d)]$, which can then be factored out of the brackets. Since the approximation is concerned with the $W_H(k)$ for k near m, $\sin\,\pi\frac{(k-m-d\pm1}{N})$ is approximately equal to $\pi\frac{(k-m-d\pm1)}{N}$ for reasonably large N, say on the order of $10^3$. Similarly the factor $(\frac{N-1}{N})$ appearing in the exponentials is approximately unity for N large. Thus, for k in the vicinity of m,

$$W_H(k) \approx \frac{1}{2N} e^{j\alpha} \quad e^{j\pi d} \; e^{j\pi(k-m)} \; \sin[\pi(k-m-d)]\frac{N}{\pi}\left\{\frac{1}{(k-m-d)} + \frac{1}{2(k-m-d+1)} + \frac{1}{2(k-m-d-1)}\right\} \qquad 3.22$$

Factoring a $\frac{1}{(k-m-d)}$ from inside the brackets and combining,

$$W_H(k) \approx \frac{1}{2}\, e^{j(\alpha+\pi d)} \cdot (-1)^{(k-m)}\left\{\frac{\sin\pi(k-m-d)}{\pi(k-m-d)} \cdot \left(\frac{2(k-m-d)^2-1}{(k-m-d)^2-1}\right)\right\}$$

3.23

The magnitudes for the frequency contributions around $k = m$ are given by the factor in brackets. The phases are determined by $\alpha + \pi d$, plus some number of $\pi$ rotations determined by $k - m$ in the last exponential, and the sign of the factor in brackets. In the example given below, it is shown that these factors of $(-1)^n$ are responsible for three consecutive phase shifts of $\pi$ radians each in the vicinity of $k = m$.

The approximate $W_H(k)$ given by 3.23 to be generated for the synthesis algorithm has the following exploitable characteristic. As was noted before, only the phase $\alpha$ changes for any of the $W_H(k)$ from one segment to the next. Furthermore, the only phases required are either $\alpha + \pi d$ or $\alpha + \pi d + \pi$. Thus, if M magnitudes are computed and stored for an M cell approximation, only one cosine and one sine are needed to determine the real and imaginary parts to be added to the synthesized spectrum for each segment. Note that unless a particular phase is required, the initial phase ($\alpha+\pi d$) for the first segment can be arbitrary, and all that is required thereafter is to increase the $\alpha+\pi d$ term by $\pi d$ on each iteration.

Since the primary virture of Hanning weighting is to greatly reduce the sidelobes of the sinc(x) DFT filter function, it provides almost the same reduction in the tails of the $\text{sind}_N(x)$ form of 3.8. The result is that, although as before perfect synthesis requires contributions in all frequency cells, the Hanning modulated discrete segment achieves a very close approximation with many fewer non-zero components.

The basic parameters of the previous example were repeated in the example below illustrating the application of the "overlap-Hanned" algorithm. A set of time function segments, each consisting of 10.25 cycles in 128 points, with phases incremented by $2\pi/8$ rad. for each successive segment, are forward transformed, Hanned, tails truncated, inverse transformed and combined to form a composite output time function of 41 cycles in 512 points as before. Figure 3.9 shows the same 10.25 cycle segment as shown in 3.1, but after modulation by the raised cosine pulse, of "Hanning". The corresponding spectrum is shown in Figure 3.10. Note that the tails of the spectrum have been reduced considerably, such that barely four components on each side of the actual

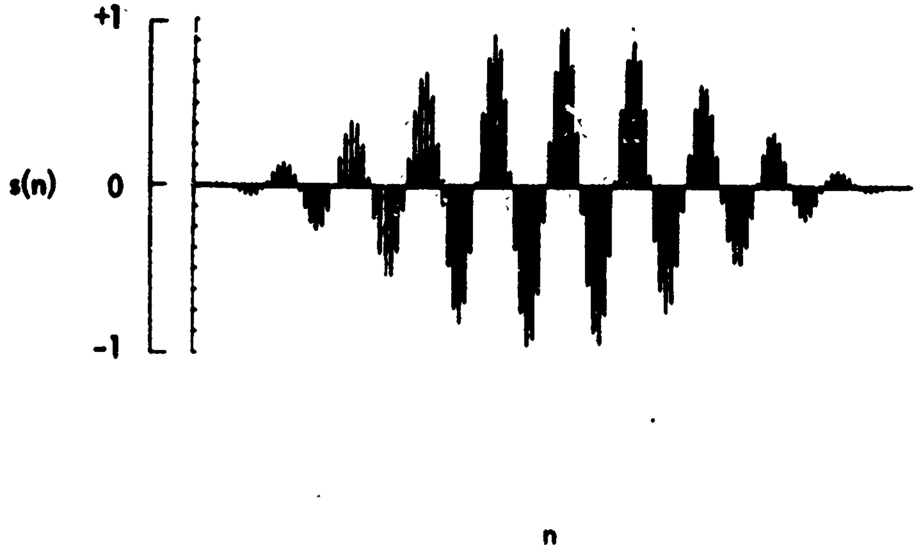


FIG. 3.9 HANNED SINUSOID SEGMENT - 10.25 CYCLES, 128 SAMPLES.

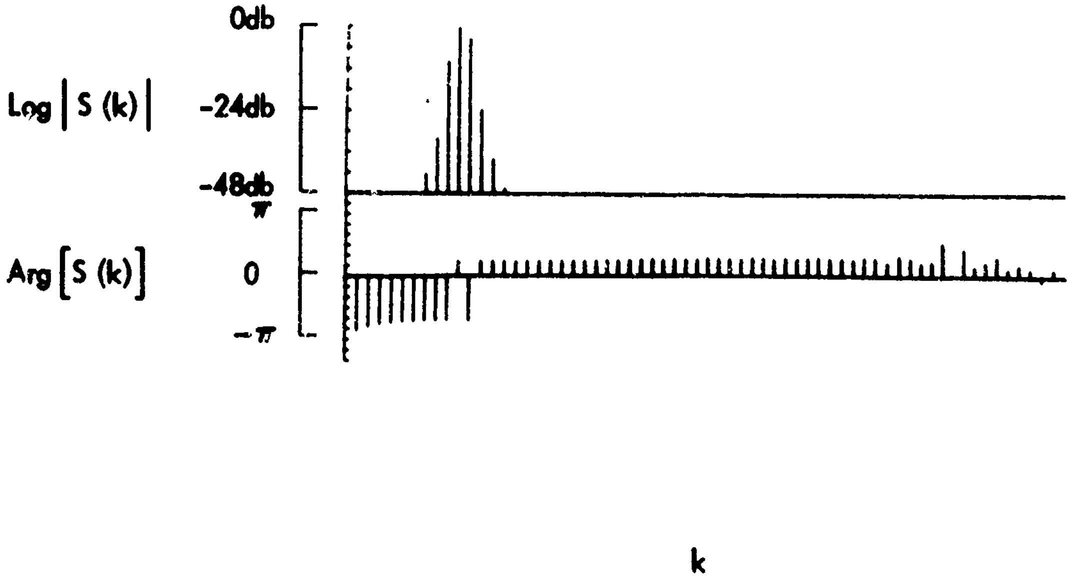


FIG. 3.10 SPECTRUM OF HANNED 10.25 CYCLE SEGMENT.

frequency are within the 48 db range plotted. Note in addition that the phase undergoes three shifts of $\pi$ radians each in the immediate vicinity of the desired frequency, compared to the single $\pi$ radian shift (Figure 3.2) for the unHanned segment.

The tails of the spectrum of each Hanned segment are then truncated to produce the spectrum of the approximation as before. For this example, only the six largest components in Figure 3.9 are retained, which corresponds to throwing away all contributions that are more than about 40 db below the largest one. It is interesting to note that, in Figure 3.2, no components in the unHanned segment are below the -40 db level. Finally Figure 3.11 shows the composite 512 sample approximation waveform obtained by the overlap-Hanned algorithm. The error is plotted in Figure 3.12. The approximation spectrum, shown in Figure 3.13 has no sidelobe components larger than 50 db below the desired component. Figure 3.14 is the spectrum of the error alone, which consists primarily of components displaced by multiples of $2/NT_s$, or the reciprocal of half the segment length. The doubling of the effective modulation rate relative to the first algorithm is due to the factor of 2 overlap, since the modulation of each segment is definitely still characterized by the period $NT_s$.

The "overlap-Hanned" synthesis algorithm for discrete components yields very high fidelity with a very small

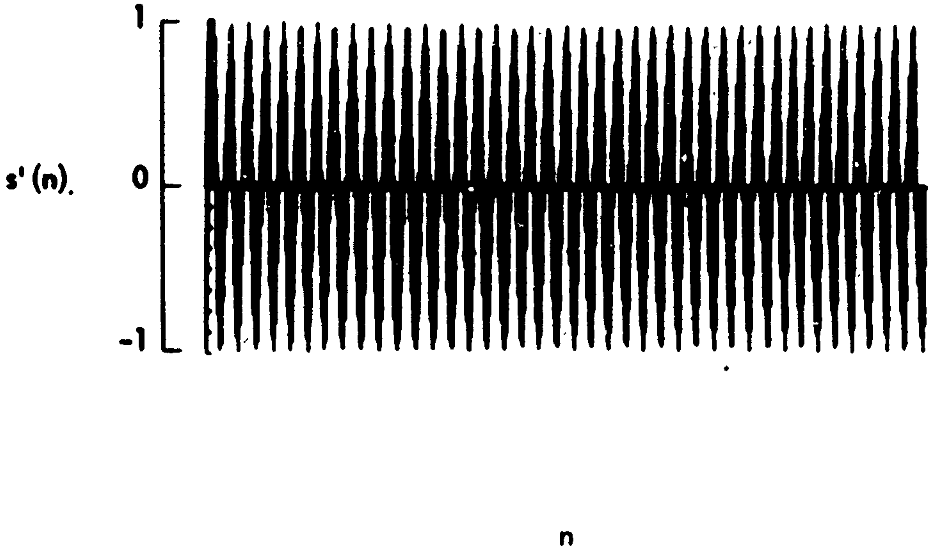


FIG. 3.11 COMPOSITE APPROXIMATION FROM 8 HANNED SEGMENTS.

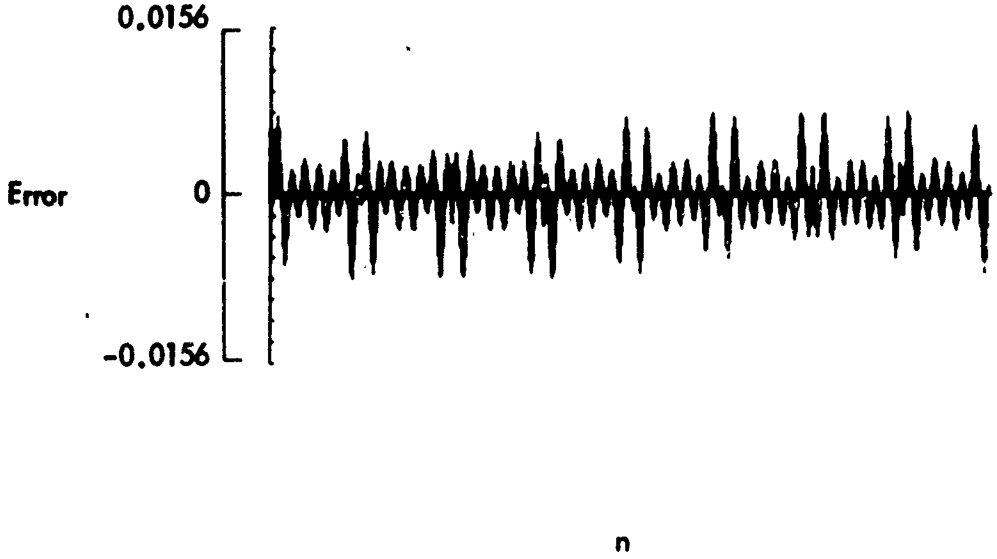


FIG. 3.12 ERROR s'(n)-s(n) FOR OVERLAP-HANNED APPROXIMATION.

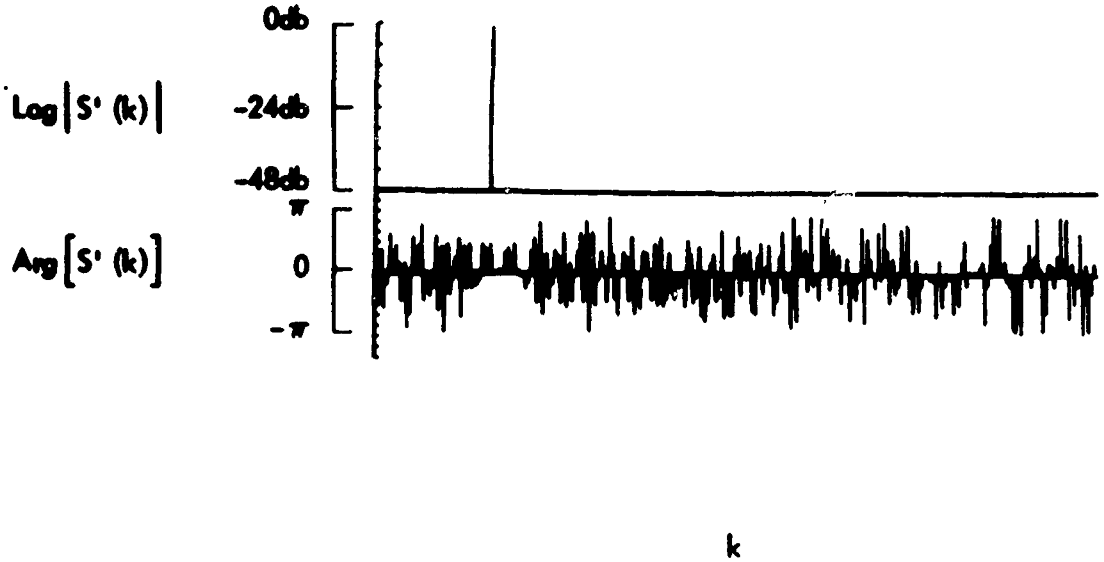


FIG. 3.13 SPECTRUM OF OVERLAP-HANNED APPROXIMATION.

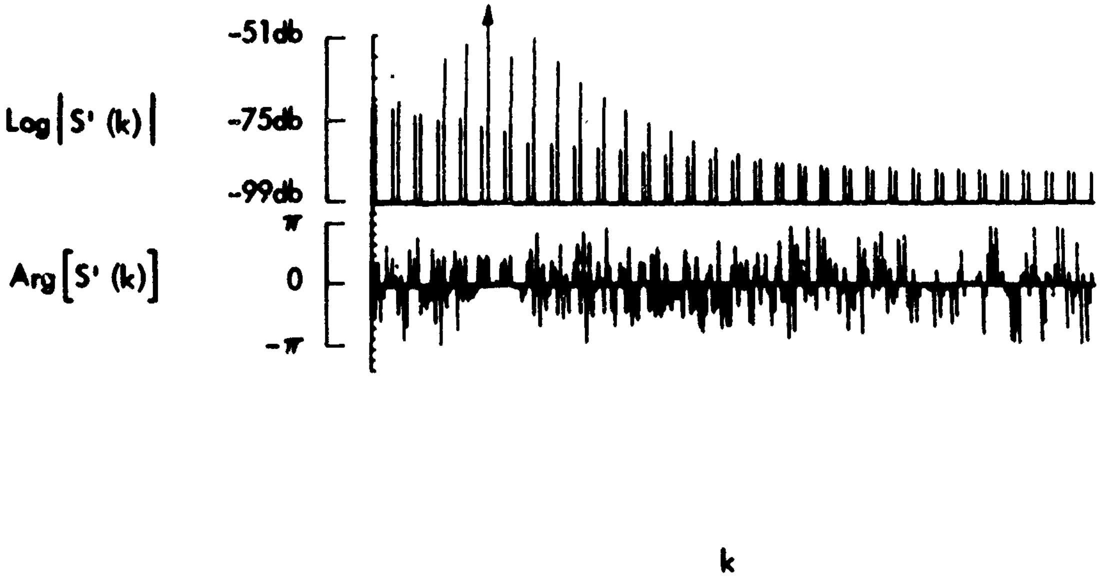


FIG. 3.14 SPECTRUM OF OVERLAP-HANNED APPROXIMATION - RESCALED.

number of non-zero spectral contributions. In particular, a six component approximation empirically appears to yield a maximum error of slightly more than 1%, and maximum spectral sidelobes of about -50 db. Even a 4 cell approximation has less than 5% error, and about -35 db max sidelobe levels.

# CHAPTER IV

## TIME DOMAIN SYNTHESIS OF DISCRETE FREQUENCIES

### A. Harmonic Sets

Quite often it is desirable to generate sets of narrowband components which are harmonically related. That is, all of the members of a particular set are just the spectral components $Y_k$ of a periodic or almost periodic time waveform. Let the time function of a particular harmonic set be denoted by y(t), defined by

$$y(t) = \sum_{k=-\infty}^{\infty} Y_k \quad e^{j2\pi kf_o t} \qquad 4.1$$

where $f_o$ is the fundamental frequency of y(t). The spectrum of y(t) is then

$$Y(f) = \sum_{k=-\infty}^{\infty} Y_k \quad \delta(f - kf_o) \quad . \qquad 4.2$$

Although each of the components of Y(f) could be generated explicitly by the techniques described in Chapter 3, it becomes profitable to generate y(t) directly in the time domain if the number of non-zero spectral components $Y(kf_o)$ is fairly large.

If identical copies of a waveform defined over an interval $T_o$ are concatenated in time, the result is a periodic signal with fundamental frequency $1/T_o$. Thus if a lossless delay line of length $T_o$ containing one period of a desired waveform were recirculated as illustrated in Figure 4.1 the signal observable at any tap would be the periodic signal desired, with fundamental frequency $f_o = 1/T_o$. If the waveform contained in the delay line is exactly one full cycle of a sinusoid, the spectrum of the signal generated will be non-zero only at the fundamental frequency $\pm f_o$. However, if an arbitrary waveform is contained in the delay line, then in general the spectrum may contain any of the frequencies harmonically related to $f_o$, i.e., $kf_o$ for all integer k. The spectral values at these harmonic frequencies are just the coefficients obtained by expanding the $T_o$ seconds of signal in the delay line in a Fourier series.

The discrete equivalent of the above operation is obtained by loading a shift register with the samples of one period of a waveform y(t), connecting the last stage to the input for recirculation, and observing a discrete periodic signal $y^x(t)$ at any stage in the shift register. For an N stage register and a shift rate $f_r$, the sampled periodic signal produced has a fundamental frequency $f_o = f_r/N$. Since the resultant waveform is both sampled and periodic, its spectrum $Y^x(f)$ is also both sampled and

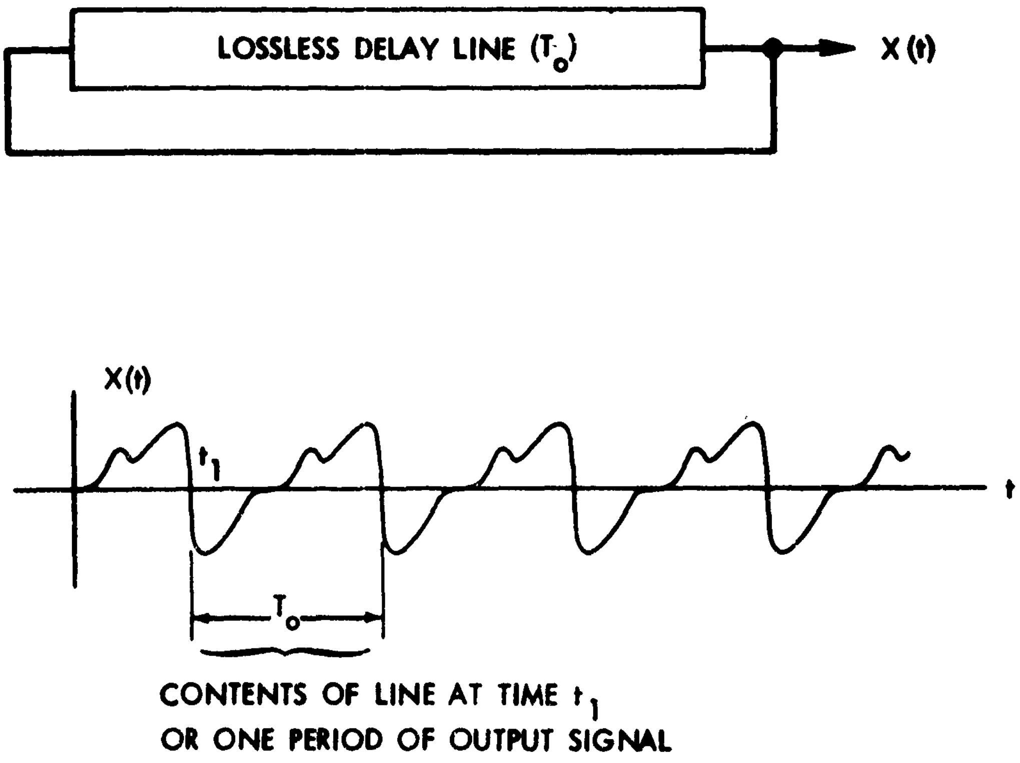

LOSSLESS DELAY LINE ($T_o$)
X(t)
X(t)
$t_1$
t
$T_o$
CONTENTS OF LINE AT TIME $t_1$
OR ONE PERIOD OF OUTPUT SIGNAL

FIG. 4.1 GENERATION OF PERIODIC SIGNAL BY RECIRCULATING DELAY LINE

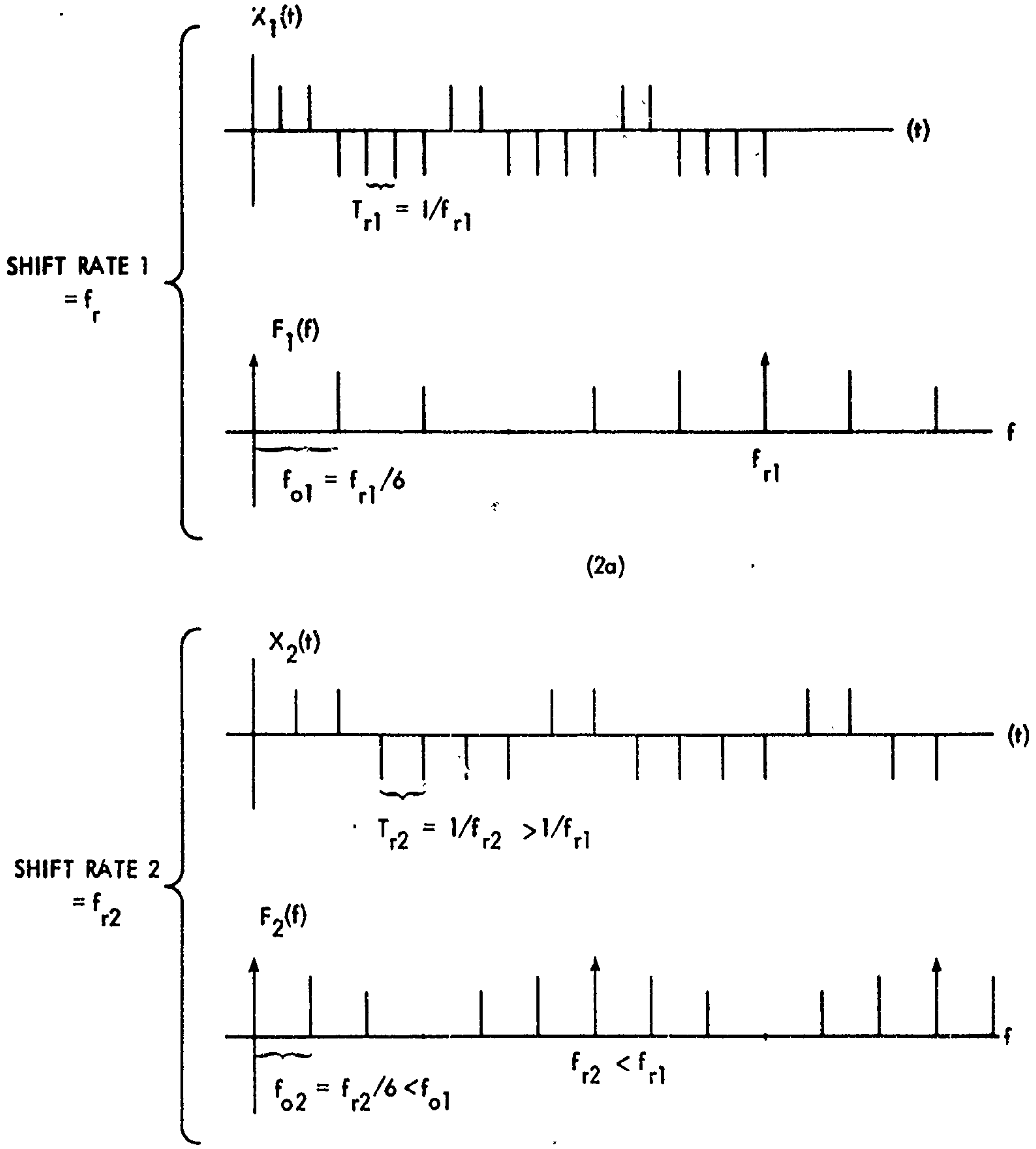
periodic, with a sample spacing of $f_o$ and period $Nf_o = f_r$. Since a shift register may be shifted at a variety of rates, the fundamental frequency of the signal generated by the discrete system may be easily adjusted to any desired value by the appropriate choice of shift rate $f_r$. Figure 4.2 illustrates the output of a six state register for two different shift rates, and the corresponding spectra for each.

If the shift register version described above is applied to the generation of a periodic signal with an arbitrary harmonic structure, care must be taken that the shift register is sufficiently long to allow independent definition of all desired harmonics. Since the discrete spectrum $Y^x(f)$ is periodic with period $Nf_o$, it is obvious that there are at most N different complex harmonic amplitudes. However, since $y^x(t)$ is a real time series, the spectral components exhibit conjugate symmetry about $f = 0$; i.e., if

$$y^x(t) = \sum_{k=-\infty}^{\infty} Y_k \quad e^{\frac{j2\pi kt}{T_o}} = \text{Real} \qquad 4.3$$

then

$$Y_k = Y^*_{-k} \qquad 4.4$$



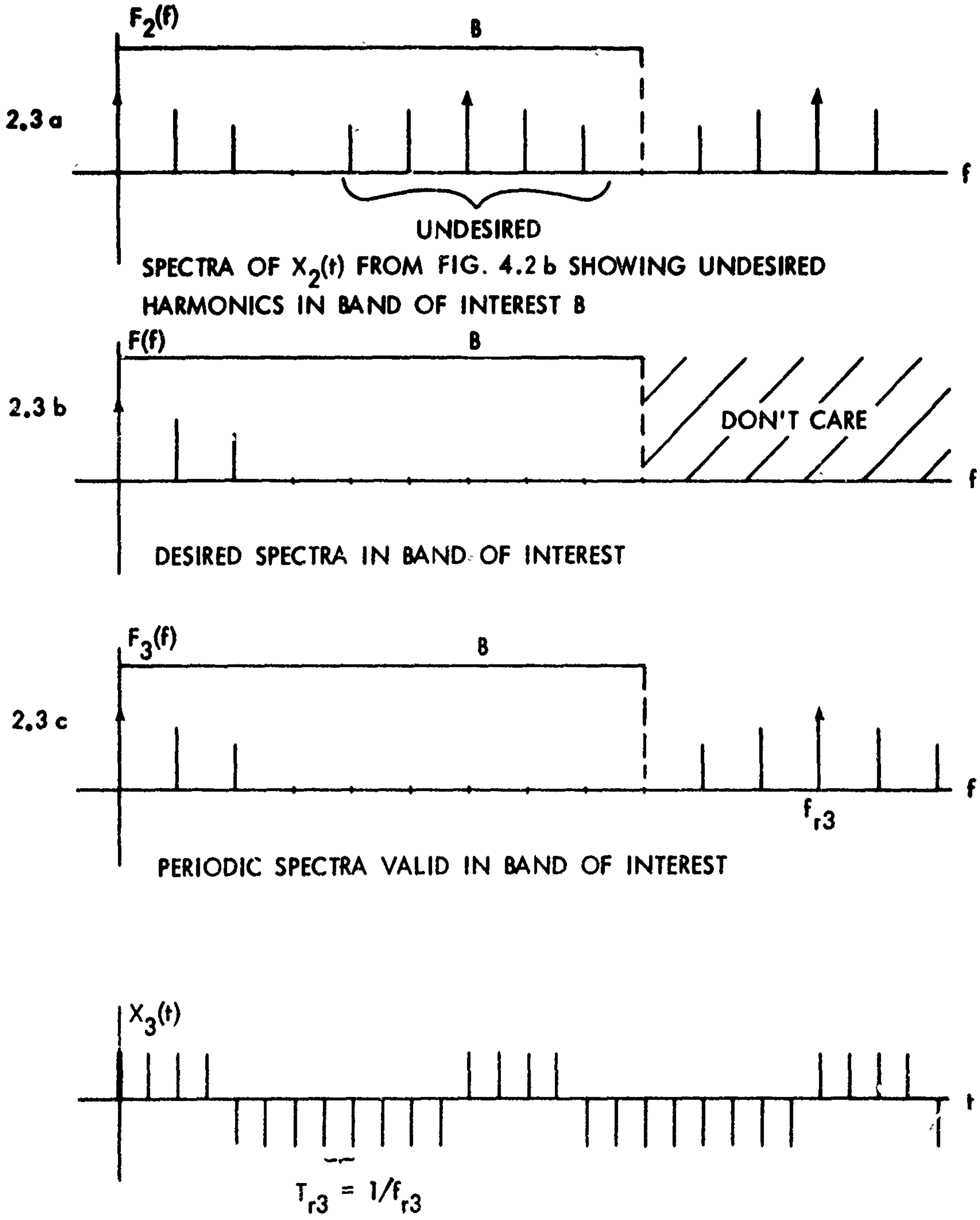
(2a)

(2b)

FIG. 4.2 OUTPUT OF 6 STAGE SHIFT REGISTER FOR TWO DIFFERENT SHIFT RATES

where * denotes the complex conjugate. Thus for a N stage shift register, at most [(N-1)/2] harmonics plus the d.c. component are independently specifiable, where [ ] denotes the integer part.

A further complication exists when the valid output band may be significantly wider than the band covered by the desired harmonics of the generated signal. Figure 4.3 illustrates a case where, if zero filled harmonics are not explicitly included in the waveform stored in the shift register (corresponding to sampling at greater than the Nyquist rate and therefore additional length of the shift register), undesired harmonics are created in the output band of interest for low shift rates (low fundamental frequencies) due to the periodic nature of the generated signal spectrum. In general, if the required valid output bandwidth is B and the lowest fundamental frequency required is $f_{min}$, the minimum number of harmonics that must be specified is $B/f_{min}$, or equivalently, the minimum shift register length is $2B/f_{min}$. From Figure 4.3c it can be seen that if the highest non-zero desired harmonic is m, then the minimum value the shift rate can be is $B + mf_{min}$, or the valid output band plus room for the image spectra at the shift rate $f_r$. The minimum length of the shift register is then given by the next integer larger than $f_r/f_{min}$, i.e., $N_{min} = [f_r/f_{min}]+1 = [B/f_{min}]+m+1$. Thus, for all fundamental frequencies for which the



SPECTRA OF $X_2(t)$ FROM FIG. 4.2 b SHOWING UNDESIRED HARMONICS IN BAND OF INTEREST B

DESIRED SPECTRA IN BAND OF INTEREST

PERIODIC SPECTRA VALID IN BAND OF INTEREST

CORRESPONDING SHIFT REGISTER OUTPUT REQUIRES 12 STAGE SHIFT REGISTER

FIG. 4.3 MINIMUM REGISTER LENGTH FOR GUARD BAND

desired waveform must be generated, the unwanted harmonics are outside the band of interest and may be removed by analog filtering after D/A conversion, or just ignored.

### B. The "Slip-Sample" Algorithm

The above technique for generating periodic signals with arbitrary harmonic structure and variable fundamental frequency requires a dedicated recirculating shift register and a controllable rate shift clock, both of which are special purpose hardware. The following technique is an adaptation of the principle of the above implementation, but is designed to be realized as software in a general purpose computer.

Assume that the required number of discrete samples of one period of the desired signal is stored in a block of random access computer memory. If the samples are sequentially read from the block by the computer and fed to the output D/A converters, the result is identical to the shift register output described above (the computer returns to the beginning of the block after each pass through the block). The rate at which the samples are read and output is determined by a clock interrupt or other timing technique, and as before determines the fundamental frequency of the output signal.

In a more general case, however, the generation of the periodic signal may be only a part of the overall signal processing problem, and may be fed directly to other parts of the sampled data system. In the usual case the entire system will be operating at one fixed sample rate $f_s$, and each part of the system must accept and process data at the rate $f_s$. This rate is not in general related to the desired fundamental frequency $f_o$ of the periodic signal to be generated.

One possible solution to this constrained problem would be to adjust the size of the memory block to be such that at the sample rate $f_s$, once through the block would be $T_o = 1/f_o$ seconds, where $f_o$ is the desired fundamental. This has two rather severe drawbacks. First, the realizable fundamental frequencies are given by $f_s/n$, where n is an integer. Thus, except for extremely low fundamental frequencies (large n), the realizable frequencies are few and far between. In addition, even for very low fundamentals where reasonable frequency resolution is available, each change in fundamental frequency requires rederiving the complete set of time samples stored in the memory block.

A more viable solution to the fixed sample rate problem is to maintain a fixed block size adequate for the lowest desired fundamental frequency, and to vary the effective rate at which the computer scans through the block. This variation in readout rate is accomplished by maintaining and incrementing the position in the memory block with a precision much greater than one memory cell. Since it is not really possible to read between memory cells or desirable for computational loading considerations to interpolate, the block position is rounded (or truncated) to the nearest integer memory location for the actual data access. In simple terms, the net effect is to "stretch" the stored waveform by occasionally duplicating a value in the output (reading a memory cell twice before going on) or to "shrink" the waveform by occasionally skipping a cell in the readout.

Consider a memory block of length N and block position parameter P, as shown in Figure 4.4. Assume that P is maintained to a sufficiently high resolution as to be essentially continuous relative to the quantization of the memory into N words. If the block of memory contains one period of some waveform, then P is in effect a phase angle of the fundamental of the periodic waveform. For a non-integer P between the 4th and 5th samples as illustrated, the contents of the fourth cell are used as the output for, say, the $i^{th}$ sample. P is then incremented by an amount F derived from the desired fundamental frequency,

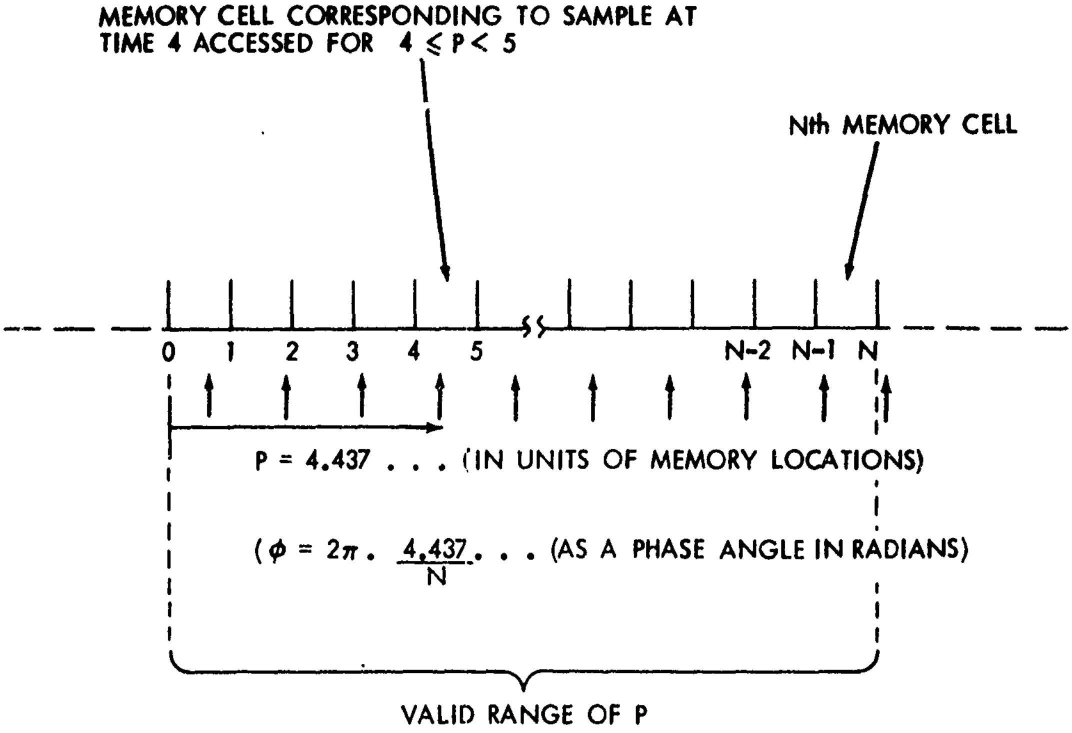

MEMORY CELL CORRESPONDING TO SAMPLE AT TIME 4 ACCESSED FOR 4 ≤ P < 5
Nth MEMORY CELL
0
1
2
3
4
5
N-2
N-1
N
P = 4.437 . . . (IN UNITS OF MEMORY LOCATIONS)
(φ = 2π . 4.437/N . . . (AS A PHASE ANGLE IN RADIANS)
VALID RANGE OF P

FIG. 4.4 SCHEMATIC REPRESENTATION OF CONTINUOUS POSITION IN MEMORY BLOCK AS PHASE ANGLE

and the appropriate memory cell is accessed for the $(i+1)^{st}$ sample. F is specified with the same resolution as P, and is therefore also not in general an integral number of memory cells. After incrementing by F, P is retained modulo N to maintain it within its valid range. The fundamental frequency $f_o$ is then given by the rate of passing completely through the block, or

$$f_o = f_s \cdot \frac{F}{N} \quad . \tag{4.5}$$

It is important to note here that the memory locations accessed twice (or skipped, as the case may be) are in general different on each pass through the block. In addition the number of "doubles" or "skips" per pass may in general alternate between two values such that the average number per pass is non-integral. Thus this technique differs significantly from just modifying the block by duplicating or dropping one or more samples. It is also obvious that for frequencies which require a non-integral number of "glitches" per pass, the output signal is not truly periodic, nor is it even composed of samples of the desired periodic signal. It is really only an approximation to the sampled version of the desired signal and on the average (in some sense) has the right behavior.

The analysis of the sampled data signal generated by this technique is aided by identifying the operations performed as components of a more familiar system. Let the desired periodic signal be $y(t)$, with period $T_o$ and fundamental frequency $f_o = 1/T_o$. Assume that samples are taken from $y(t)$ at a rate $f_{s_1}$ which is N times the fundamental. Thus there will be exactly N samples in each period of the sampled signal $y^x(t)$. The samples (or impulses) of $y^x(t)$ are now applied to the input of a "boxcar-integrator" filter, i.e., a filter with an impulse response of a rectangular pulse of unit amplitude and duration $T_o$ centered at $t = 0$. Let the filter output be designated $y_b(t)$. If $y_b(t)$ is now resampled at a second rate $f_{s_2}$, in general not related to $f_{s_1}$, samples taken during any interval $(n-\frac{1}{2})T_o < t < (n+\frac{1}{2})T_o$ will yield the value of the sample taken from $y(t)$ at $t = nT_o$. This is exactly analogous, to the situation where samples taken from $y(t)$ are stored in a computer memory, and then accessed by rounding sample "times" to the nearest memory address, assuming here that samples are stored such that increasing time corresponds to increasing addresses. (Truncating instead of rounding to form the address would introduce a delay of half a memory cell, but would yield equivalent overall results.) The output signal is therefore the result of 1) multiplying the desired y(t) by an impulse train of frequency $f_{s_1}$, 2) convolving the result with the impulse response of the boxcar filter, and 3) resampling its output by multiplying by the impulse train of frequency $f_{s_2}$. Correspondingly, the spectrum of the output signal is obtained from the spectrum of y(t) by 1) convolving with an impulse train of spacing $f_{s_1}$ (which is some integral multiple of the fundamental frequency) as a result of the initial sampling process, 2) multiplying by $\frac{1}{f_{s_1}}$ sinc$(f/f_{s1})$, the filter function of the boxcar integrator, and 3) convolving the result with a second impulse train of spacing $f_{s_2}$.

Since for the case of interest here the signal y(t) is periodic, its spectrum Y(f) is impulsive with spacing $f_o$:

$$Y(f) = \sum_{k=-N/2+1}^{N/2-1} Y_k \, \delta(f - kf_o) \qquad 4.6$$

where Y(f) has been explicitly band-limited in anticipation of sampling at $Nf_o$. Performing the sampling operation, Y(f) must be duplicated every $f_{s_1} = Nf_o$,

$$Y^x(f) = \sum_{m=-\infty}^{\infty} Y(f-mf_{s_1}) = \sum_{m=-\infty}^{\infty} \sum_{k=-N/2+1}^{N/2-1} Y_k \, \delta(f-f_o[k+mN]) \quad 4.7$$

Multiplying by the filter function,

$$Y_b(f) = \frac{1}{Nf_o} \operatorname{sinc}(f/Nf_o) \sum_{m=-\infty}^{\infty} \sum_{k=-N/2+1}^{N/2-1} Y_k \, \delta(f-f_o[k+mN]) \quad 4.8$$

Convolving with the second impulse train for the final step, the spectrum of the output $y_b^x(t)$ is obtained:

$$Y_b^x(f) = \sum_{r=-\infty}^{\infty} Y_b(f-rf_{s_2}) \quad 4.9$$

or

$$Y_b^x(f) = \sum_{r=-\infty}^{\infty} \frac{1}{Nf_o} \operatorname{sinc}\left(\frac{f-rf_{s_2}}{Nf_o}\right) \sum_{m=-\infty}^{\infty} \sum_{k=-N/2+1}^{N/2-1} Y_k \, \delta([f-rf_{s_2}]$$

$$- [k+mN]f_o) . \quad 4.10$$

The system under consideration has a fixed system sample rate $f_s$, and thus the second sample rate $f_{s_2}$ is constrained to the value $f_{s_2} = f_s$. Now define a parameter $\alpha$ as the ratio of the first sampling rate $f_{s_1}$ to $f_s$

$$\alpha = \frac{f_{s_1}}{f_s} = \frac{Nf_o}{f_s};\ f_o = \frac{\alpha f_s}{N} \qquad 4.11$$

Substituting 4.11 into 4.10,

$$Y_b^x(f) = \sum_{r=-\infty}^{\infty} \frac{1}{\alpha f_s} \operatorname{sinc}\left(\frac{f-rf_s}{\alpha f_s}\right) \sum_{m=-\infty}^{\infty} \sum_{k=-N/2+1}^{N/2-1} Y_k\ \delta([f-rf_s] - [\tfrac{k}{N}+m]\alpha f_s) \qquad 4.12$$

The resultant spectrum $Y_b^x(f)$ has a replica of the boxcarred spectrum at each multiple of $f_s$, as is expected for any sampling operation, but with the effective frequency scale of each replica about the appropriate value of $rf_s$ variable with $\alpha$. Since $\alpha$ is proportional to $f_o$, the frequency of the fundamental of the harmonic set generated can be controlled by controlling $\alpha$. From 4.11 $f_{s_1} = \alpha f_s$, or $\frac{1}{f_s} = \alpha\frac{1}{f_{s_1}}$, so that the spacing between samples for the second sampling process is $\alpha$ times the spacing of samples going into the boxcar filter. The latter samples are stored one to each memory cell in the computer memory, and thus $1/f_{s_1}$ in terms of memory cell spacing is unity. The spacing in memory cells between samples for the output sampling process is therefore $\alpha$. But this spacing is just what was previously defined as the quantity F, and from 4.5 and 4.11, F = $\alpha$ is easily verified.

The initial sampling process results in a non-band limited spectrum even though the original spectrum Y(f) was band-limited. The effective boxcar filtering operation reduces the amplitude of the replicas generated by the first sampling operation, but since the $\mathrm{sinc}(f/f_{s_1})$ envelope only falls off as 1/f, considerable power still remains in the undesired replicas. After the second sampling operation these reduced but non-zero replicas of the desired spectrum in the vicinity of $\pm f_{s_1}$, $\pm 2f_{s_1}$, etc., of the box-carred spectrum may be aliased back into the band of interest by the resampling operation, and in general will appear as weak, discrete frequencies not harmonically related to the desired fundamental $f_o$. These artifacts may be quite large if generated by harmonics in the original waveform with frequencies close to 1/2 of the original sample rate $f_{s_1}$. This is due to the fact that the $\mathrm{sinc}(f/f_{s_1})$ envelope of the box-carred spectrum is down only about 4db at the Nyquist frequency $f_{s_1}/2$. If the original waveform is oversampled by a factor of two such that all non-zero harmonics are below $f_{s_1}/4$, the minimum attenuation provided by the sinc envelope becomes a little better than 10db, and increases by 6db for each additional factor of 2 in oversampling (see Figure 4.5). Thus by increasing the memory required to store the waveform it is possible to make artifacts as small as desired. However, the 6db per octave roll-off of the sinc function makes this a fairly expensive solution to the problem.

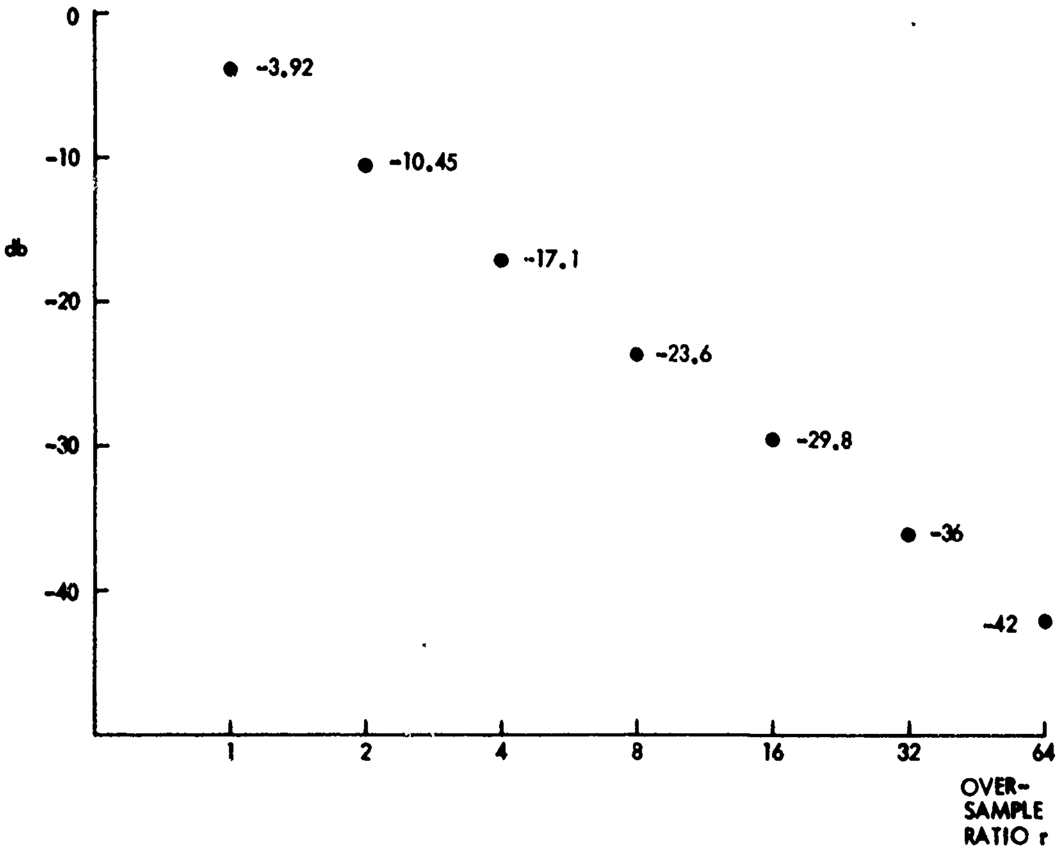


FIG. 4.5 MINIMUM ATTENUATION OF ALIASED SPECTRA FOR VARIOUS OVER-SAMPLE RATES. ($f_{S_1} = r \cdot k \cdot f_o$ ; WHERE $k \cdot f_o$ IS THE HIGHEST NON ZERO HARMONIC DESIRED)

If the memory address increment F which determines the fundamental frequency of the output is fluctuated rapidly by some small amount, the effect is to smear the power at each of the harmonically related discrete frequencies over some finite bandwidth. The bandwidth of the n$^{th}$ harmonic will be n times the bandwidth of the fundamental. In addition to being useful to generate finite bandwidth narrowband processes, this phenomena also tends to smear out the discretes in the band of interest generated by aliasing. Unfortunately if highly stable discretes are desired, the aliased contributions may still be unacceptably large.

More exotic interpolation schemes or other prefiltering operations could be performed on the stored waveform prior to resampling to reduce aliasing, but even linear interpolation requires a multiplication per sample, and therefore it as well as more sophisticated filtering schemes are impractical to implement.

### C. Stochastic Filtering

Although conventional filtering algorithms are impractical, an effective prefiltering operation can be performed by "jittered" resampling, and thus the aliasing problem can be reduced. The "jittered" sampling technique is quite attractive relative to more straightforward prefiltering methods due to its extremely simple implementation. Practical application to the Memory-Read-Out signal generation problem will be discussed after deriving the spectrum of the output if "jittered" sampling is employed.

The problem is as follows: the function s(t) is given with Power Spectral Density (PSD) $\Phi_s(f)$ and autocorrelation function (ACF) $\phi_s(\tau)$. Therefore

$$\phi_s(\tau) = \overline{s(t)\, s(t+\tau)} \qquad 4.13$$

and

$$\Phi_s(f) = \int_{-\infty}^{\infty} \phi_s(\tau)\, e^{-j2\pi f\tau} d\tau \qquad 4.14$$

is just the Fourier Transform of $\phi_s(\tau)$.

Samples are now taken of s(t) according to the "jittered" sampling scheme; i.e., the $i^{th}$ sample is taken at the time $i\,T_s + \nu_i$, where $T_s = 1/f_s$ is the period associated with the underlying basic sample rate $f_s$, and the $\nu_i$ are independent, identically distributed random variables with probability density function (PDF)

$$p_{\nu_i}(t) = p(t)\ , \quad \text{all } i \qquad 4.15$$

The sample values of s(t) obtained are then assigned to a regularly spaced impulse train of period $T_s$ to form a sampled function $s^x(t)$ given by

$$\underset{\sim}{s}^x(t) = \sum_{n=-\infty}^{\infty} s(nT_s + \underset{\sim}{\nu}_n) \quad \delta(t - nT_s) \qquad 4.16$$

The problem thus is to determine the PSD of $\underset{\sim}{s}^x(t)$ in terms of $\phi_s(f)$, the PSD of $s(t)$; and of $p(t)$, the PDF of the $\underset{\sim}{\nu}_i$.

Consider the sequence autocorrelation of the original "jittered" samples of $s(t)$:

$$R(r) = \overline{s(nT_s + \underset{\sim}{\nu}_n)s((n+r)T_s + \underset{\sim}{\nu}_{n+r})} \qquad 4.17$$

where the average is to be taken over all n and over the $\nu_i$. For $\underset{\sim}{\nu}_n$ and $\underset{\sim}{\nu}_{n+r}$ identically equal to zero, $R(r)$ is just the autocorrelation of $s(t)$ for $rT_s$ delay, i.e.,

$$R(r) = \phi_s(rT_s); \qquad \text{all } \nu \equiv 0 \qquad 4.18$$

For an arbitrary distribution function $p(t)$ for the $\nu_i$, $R(r)$ is the statistical average of $\phi_s(\tau)$ taken over all possible $\underset{\sim}{\nu}_{n+r} - \underset{\sim}{\nu}_n$. For $r \neq 0$ the $\underset{\sim}{\nu}_i$ are independent, and the PDF of the difference of two of the $\underset{\sim}{\nu}_i$ is just the auto-correlation of the PDF of each. Let $p_d(t)$ be the PDF of $\underset{\sim}{\nu}_{n+r} - \underset{\sim}{\nu}_n$, i.e.,

$$p_d(t) = \int_{-\infty}^{\infty} p(\sigma)\, p(\sigma+t)d\sigma \qquad 4.19$$

Then the desired sequence autocorrelation function R(r) is given by:

$$R(r) = \int_{-\infty}^{\infty} p_d(t)\ \phi_s(rT_s + t)dt; \quad r \neq 0\ . \qquad 4.20$$

Since $p_d(t)$ is real, positive and symmetric, 4.20 may equivalently be written

$$R(r) = \int_{-\infty}^{\infty} p_d(t)\ \phi_s(rT_s - t)dt; \quad r \neq 0\ . \qquad 4.21$$

If the values of R(r) are now assigned to a regularl spaced impulse train of period $T_s$, the discrete autocorrelation function of the sampled function $\tilde{s}^x(t)$ is obtained. Let $R^x(\tau)$ be the autocorrelation function of $\tilde{s}^x(t)$ . Then

$$R^x(\tau) = \sum_{\substack{r=-\infty \\ r\neq 0}}^{\infty} R(r)\ \ \delta(\tau - rT_s) + R^x(0)\ . \qquad 4.22$$

$R^x(0)$ is unaffected by the jittering operation and therefore must be handled as a separate case. Substituting for R(r)

$$R^x(\tau) = \sum_{\substack{r=-\infty \\ r\neq 0}}^{\infty} \left[\int_{-\infty}^{\infty} p_d(t)\phi_s(rT_s - t)dt\right]\delta(\tau - rT_s) + R^x(0) \qquad 4.23$$

Now define a continuous autocorrelation function $R(\tau)$ as follows:

$$R(\tau) = \int_{-\infty}^{\infty} p_d(t)\phi_s(\tau-t)dt \ . \qquad 4.24$$

If $R(\tau)$ is sampled at points $rT_s$, the result is $R^x(\tau)$ except for r=o. Let the Fourier transform of $R(\tau)$ be $\phi_R(f)$, and the Fourier transform of $p_d(t)$ be $\underset{\sim}{P}_d(f)$. Then from 4.24

$$\phi_R(f) = \underset{\sim}{P}_d(f)\phi_s(f) \ . \qquad 4.25$$

Finally, since $R^x(\tau)$ is (except for $\tau$=o) just a sampled version of $R(\tau)$, the Fourier transform of $R^x(\tau)$ is given by

$$\phi_{R^x}(f) = \sum_{r=-\infty}^{\infty} \phi_R(f-rf_s) + K = K + \sum_{r=-\infty}^{\infty} \underset{\sim}{P}_d(f-rf_s)\phi_s(f-rf_s) \qquad 4.26$$

The constant K arises due to the fact that $R^x(0) \neq R(0)$ since

$$R^x(0) = E\{[S(nT_s + \underset{\sim}{\nu}_i)]^2\} = \phi_s(0) \qquad 4.27$$

and

$$R(0) = \int_{-\infty}^{\infty} p_d(t)\phi_s(t)dt \qquad 4.28$$

which is equal to $\phi_s(0)$ only if $p_d(t) = \delta(t)$. Thus $R^x(\tau)$ can be written

$$R^x(\tau) = \sum_{r=-\infty}^{\infty} R(\tau)\ \delta(\tau - rT_s) + [\phi_s(0) - R(0)]\ \delta(\tau). \qquad 4.29$$

The transform of the last term in 4.29 yields the constant K

$$K = \phi_s(0) - R(0)\ , \qquad 4.30$$

or, using 4.28,

$$K = \phi_s(0) - \int_{-\infty}^{\infty} p_d(t)\ \phi_s(t)\ dt\ . \qquad 4.31$$

The effect of the "jittered" sampling can be seen from 4.25 and 4.26 to be almost equivalent to passing s(t) through a filter with an impulse response equal to the PDF of the $\nu_i$, p(t). However, the one anomaly of stochastic filtering is the white noise term K which is added to the PSD of the final output. The significance of the white noise term is apparent if one realizes that the stochastic filtering is a power lossless operation, and that the power missing from the attenuated frequencies shows up as white noise. Since the objective was to reduce the discretes generated in the band of interest by aliasing, trading them off for white noise is usually quite acceptable. Further properties of "jittered" sampling are discussed in Balakirshnan(2) and Shapiro and Silverman(3).

The implementation of the stochastic filter by jittered sampling in the memory read-out algorithm is fairly simple compared with the computational requirement of normal filtering methods, assuming that the $\underset{\sim}{\nu_i}$ are readily available. After the block position or phase angle has been determined as before and saved, the $\underset{\sim}{\nu_i}$ is added to the phase and the result taken modulo N to determine the memory cell to be accessed.

As a simple example, let $p_{\underset{\sim}{\nu_i}}(t)$ be uniform over one sample period $T_s$. The transform of a pulse of width $T_s$ is the familiar $T_s$ sinc$(fT_s)$. Since $p_d(t)$ is the autocorrelation function of $p_{\nu_i}(t)$, $\underline{P}_d(f)$ becomes

$$\underline{P}_d(f) = T_s^2 \, \mathrm{sinc}^2(fT_s) \qquad 4.32$$

This has its first zero at $f = f_s = 1/T_s$. Since the power spectrum of the signal to be resampled was shown to be already reduced by a sinc$^2(fT_s)$ envelope due to an effective "box car" filter, the final spectrum before resampling has been reduced by sinc$^4(fT_s)$ with uniform jittered sampling. More desirable filter functions may be obtainable by careful design of the $\underset{\sim}{v}_i$ density function, but the practical advantage of the technique would soon disappear.

In summary of Chapter 4 then, a technique for generating harmonically related narrowband processes directly in the time domain has been described. The technique is designed to be implemented in a digital system where the output must be constrainedto signal samples at some system-determined sample rate. The spectrum of the resulting generated signal was described and shown to have potentially severe aliasing problems. Increased memory for waveform storage, interpolation schemes, and finally a statistical filtering technique based on "jittered" sampling are discussed as means of reducing the aliasing to tolerable levels.

# CHAPTER V

## SUMMARY AND CONCLUSION

In summary, several algorithms for the generation of steady-state signal components with all-digital systems have been presented and analyzed. Two abbreviated IFFT algorithms for the synthesis of a broad-band random process having a controllable Power Spectral Density function were analyzed and compared to a more conventional approach of passing white noise through a Finite Impulse Response (FIR) filter. The simpler of the two algorithms involves concatenating segments of signal having an appropriate spectral composition, and results in a savings of 75% of the computational effort required in the FIR approach. However, the synthesized signal has a discontinuity at each segment boundary which may restrict its application. The second algorithm employs summing an overlap of appropriately weighted segments to eliminate the discontinuity, although the derivatives at the segment ends are still discontinuous. The overlap version results in a 50% savings in computation load over the FIR approach.

For the synthesis of discrete components and very narrow-band processes, a pair of algorithms for generating approximation segments of the desired signal by inverse FFT were investigated. Here the simplest version employing no overlap in output segments gives relatively poor results. On the other hand, a corresponding version employing the sum of appropriately weighted and overlapped segments gives very good results with a minimal computational effort. The two overlap algorithms, one for broadband components and one for narrowband or discrete components, may be combined to produce efficiently with an FFT based system a signal having both types of components.

Finally, an algorithm for producing harmonically rich, periodic signals in the time domain was developed and analyzed and shown to have potentially severe aliasing problems. Various techniques to minimize the undesirable effects of the aliasing are discussed. The result is an efficient, highly implementable algorithm for generating harmonically related signal components.

The synthesis of signals with specific controllable characteristics is a relatively new and still developing part of the signal processing field. As the quality and sophistication of signal generation systems improves, more and more applications are found for their outputs. Especially with all digital systems, the ability to produce signals modeling problems with random processes, and yet to be able to reproduce the synthetic signal exactly or with a controlled perturbation is extremely useful in evaluating the performance of a signal analysis system. The application to simulators, particularly in training devices, is readily apparent. However, although many new doors are opened by the advent of practical, all-digital signal synthesis, the results of this paper clearly indicate the need for careful analysis in applying the various techniques to any specific problem.

Several of the results in the paper have potential application in other areas of digital signal processing. The technique discussed in Chapter 3 of summing overlapped time blocks which have been derived by inverse transforming Hanning weighted spectra forms the basis for an extremely flexible FFT filtering algorithm. The results found in Chapter 4 for the spectrum of a resampled digital signal stored in a delay line or memory is of interest, for example, in a system where a sampled signal passing through a shift register delay line is observed by a tap, the position of which is changing with time. Also in Chapter 4, the concept of using randomly disturbed sampling times to accomplish at least rudimentary filtering as an aid in minimizing aliasing would seem useful, although over a decade has passed since Balakrishnan's paper and little or no application has been made of which the author is aware.

An effort to determine what useful filter functions might be obtainable under the constraints imposed on its Fourier transform would probably be worthwhile.

# APPENDIX A

Consider a truncated summation of the form

$$S = \sum_{n=0}^{N-1} \lambda^{n} \qquad \text{(A1)}$$

Manipulating,

$$\lambda S = \sum_{n=1}^{N} \lambda^{n} = \lambda^{N} + S - 1 \qquad \text{(A2)}$$

$$S - \lambda S = 1 - \lambda^{N}. \qquad \text{(A3)}$$

or

$$S = \frac{1 - \lambda^{N}}{1 - \lambda} \qquad \text{(A4)}$$

Suppose λ is a complex exponential such that S is of the form

$$S = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}nu} \qquad \text{(A5)}$$

Applying A4, with $\lambda = e^{-j\frac{2\pi}{N}u}$

$$S = \frac{1 - e^{-j2\pi u}}{1 - e^{-j\frac{2\pi}{N}u}} \tag{A6}$$

Factoring an $e^{-j\pi u}$ from the numerator and an $e^{-j\frac{\pi}{N}u}$ from the denominator,

$$S = \frac{e^{-j\pi u}}{e^{-j\frac{\pi}{N}u}} \left( \frac{e^{j\pi u} - e^{-j\pi u}}{e^{j\frac{\pi}{N}u} - e^{-j\frac{\pi}{N}u}} \right) \tag{A7}$$

or

$$S = e^{-j\pi\left(\frac{N-1}{N}\right)u} \left(\frac{\sin \pi u}{\sin \frac{\pi}{N}u}\right) \tag{A8}$$

Using the definition

$$\operatorname{sind}_N(u) = \frac{1}{N} \frac{\sin \pi u}{\sin \frac{\pi}{N}u} \tag{A9}$$

the desired result is

$$\boxed{\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}nu} = Ne^{-j\pi\left(\frac{N-1}{N}\right)u} \operatorname{sind}_N(u)} \tag{A10}$$

The $\text{sind}_N(u)$ function is plotted in Figure A1 for $N=9$. Note that for u equal to integral multiples of N, the sind has unit magnitude; for all other integral values of u, the sind is zero. For even N, the peaks alternate between plus and minus one; for odd N all peaks are positive unity. The $\text{sind}_N(u)$ is always periodic with period 2N; the magnitude (or square) is periodic with period N.

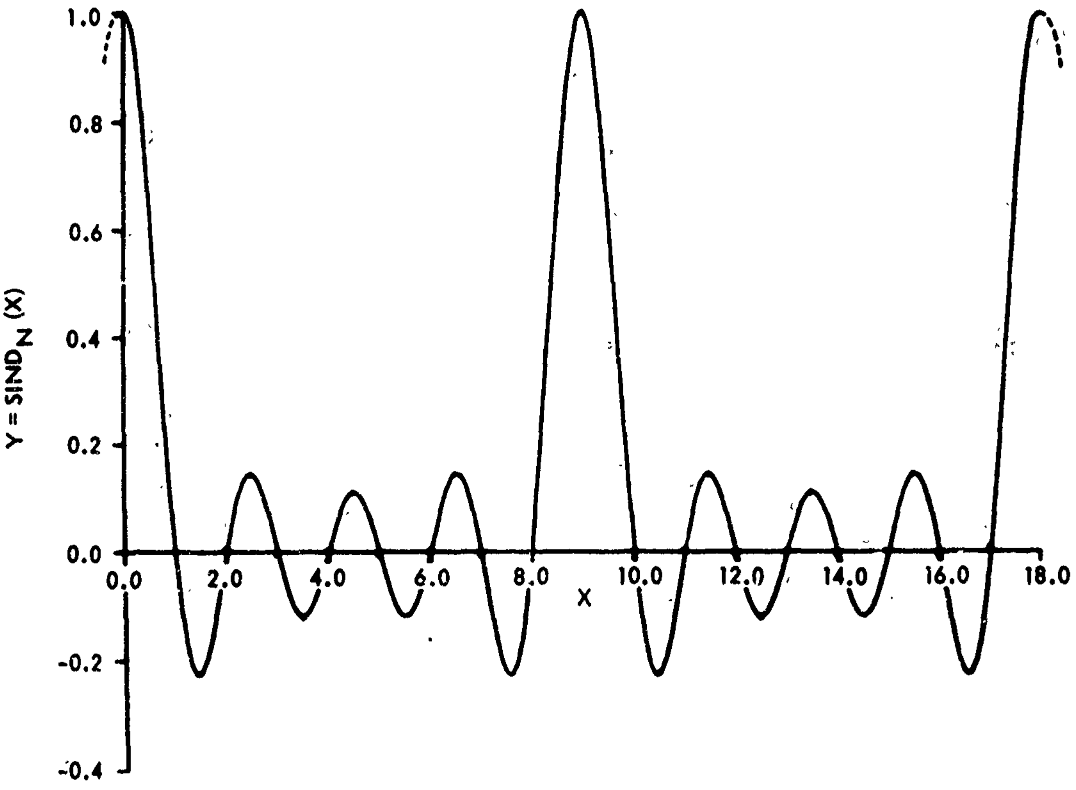


FIGURE A1 PLOT OF SIND$_N$ (X) FOR N = 9

# APPENDIX B

This appendix derives closed form expressions for the infinite series

$$S_1(u,N) = \sum_{r=-\infty}^{\infty} \operatorname{sinc}(u+rN) \tag{B1}$$

and

$$S_2(u,N) = \sum_{r=-\infty}^{\infty} \operatorname{sinc}^2(u+rN) \tag{B2}$$

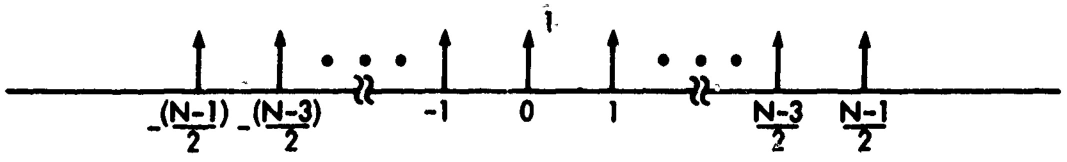
that is, the sum of a periodic series of sinc(u) functions or their squares, spaced along the u axis by a distance N.

Consider first the spectrum G(f) of a function g(t) consisting of a series of N impulses, N odd, with unit spacing and centered at t=o as shown in Figure B1. Direct computation of G(f) gives

$$G(f) = \int_{-\infty}^{\infty} g(t)\, e^{-j2\pi ft}\, dt \quad . \tag{B3}$$

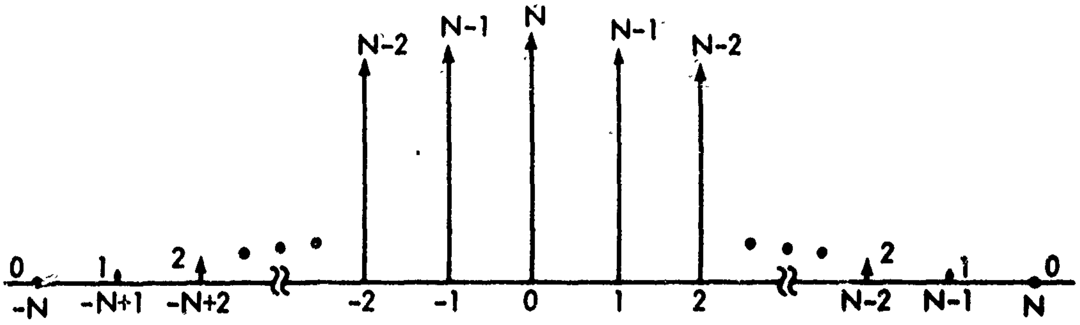
$$G(f) = \int_{-\infty}^{\infty} \sum_{n=-(N-1)/2}^{(N-1)/2} \delta(t-n)\;\; e^{-j2\pi ft}\, dt \tag{B4}$$

Interchanging the order of summation and integration



BLOCK OF N UNIT IMPLUSES, N ODD

FIGURE B1



BLOCK OF 2N-1 IMPULSES, TRIANGULAR WEIGHTING

FIGURE B2

$$G(f) = \sum_{n=-(N-1)/2}^{(N-1)/2} \int_{-\infty}^{\infty} \delta(t-n)\, e^{-j2\pi ft}\, dt \tag{B5}$$

$$G(f) = \sum_{n=\frac{-(N-1)}{2}}^{\frac{N-1}{2}} e^{-j2\pi fn} \tag{B6}$$

Writing out the series,

$$G(f) = e^{+j2\pi f\frac{(N-1)}{2}} + e^{j2\pi f\frac{(N-2)}{2}} + \ldots + e^{-j2\pi f\frac{(N-1)}{2}} \tag{B7}$$

Factoring out an $e^{j2\pi f(\frac{N-1}{2})}$,

$$G(f) = e^{j\pi f(N-1)} \sum_{n=o}^{N-1} e^{-j2\pi fn} \tag{B8}$$

Using A10, with u = Nf

$$G(f) = e^{j\pi f(N-1)} \cdot e^{-j\pi f(N-1)} \cdot N\, \text{sind}_N(Nf) \tag{B9}$$

or

$$G(f) = N\, \text{sind}_N(Nf) \tag{B10}$$

Now consider the same function g(t), but derived as an infinite impulse train

$$g'(t) = \sum_{n=-\infty}^{\infty} \delta(t+n) \tag{B11}$$

multiplied by a unit pulse of width N

$$p(t) = \begin{cases} 1 \; ; & |t| \leq \frac{N}{2} \\ 0 \; ; & |t| > \frac{N}{2} \end{cases} \tag{B12}$$

The spectrum of the impulse train is[4]

$$G'(f) = \sum_{r=-\infty}^{\infty} \delta(f+r) \tag{B13}$$

and the spectrum of the pulse is

$$P(f) = N\ \mathrm{sinc}(Nf) \tag{B14}$$

The spectrum of the product is the convolution of G'(f) with P(f):

$$G(f) = \int_{-\infty}^{\infty} G'(\nu)P(f-\nu)d\nu \tag{B15}$$

$$G(f) = \int_{-\infty}^{\infty} \sum_{r=-\infty}^{\infty} \delta(\nu+r)\ N \operatorname{sinc}[(f-\nu)N]d\nu \tag{B16}$$

Again interchanging the order of summation and integration,

$$G(f) = N \sum_{r=-\infty}^{\infty} \int_{-\infty}^{\infty} \operatorname{sinc}[(f-\nu)N]\delta(\nu+r)d\nu \tag{B17}$$

or

$$G(f) = N \sum_{r=-\infty}^{\infty} \operatorname{sinc}[fN+rN] \tag{B18}$$

But both B10 and B18 are expressions for the spectrum of the same function g(t), and thus must be equal. Therefore, letting $Nf = u$,

$$\boxed{\sum_{r=-\infty}^{\infty} \operatorname{sinc}(u+rN) = S_1(u,N) = \operatorname{sind}_N(u)} \tag{B19}$$

To obtain the desired relation for $S_2(u,N)$, consider the auto-convolution of g(t). By forming the auto-convolution of a series of finite pulses of width $\delta$ and height $1/\delta$, and then taking the limit as $\delta \to 0$, it can easily be seen that the auto-convolution of g(t) is a series of 2N - 1 impulses

$g_2(t)$ with a triangular weighting as illustrated in Figure B2. The spectrum of $g_2(t)$ is of course just the square of G(f),

$$G_2(f) = N^2 \mathrm{sind}_N^2(Nf) \qquad \text{(B20)}$$

But $g_2(t)$ can also be derived by multiplying the impulse train $g'(t)$ by a triangular pulse $p_2(t)$

$$p_2(t) = \begin{array}{ll} N(1 + t/N); & -N < t < 0 \\ N(1 - t/N); & 0 < t < N \\ 0 & ; |t| \geq N \end{array} \qquad \text{(B21)}$$

The spectrum of $p_2(t)$ is

$$P_2(f) = N^2 \ \mathrm{sinc}^2(Nf) \qquad \text{(B22)}$$

since the triangular pulse is just the auto convolution of the square pulse B12, and therefore has a spectrum equal to $P^2(f)$. Convolving $P_2(f)$ with $G'(f)$,

$$G_2(f) = \int_{-\infty}^{\infty} P_2(f-\nu)G'(\nu)d\nu \qquad \text{(B23)}$$

$$G_2(f) = \int_{-\infty}^{\infty} N^2 \text{ sinc}^2[(f-\nu)N] \sum_{r=-\infty}^{\infty} \delta(\nu+r)d\nu \qquad \text{(B24)}$$

Interchanging summation and integration,

$$G_2(f) = N^2 \sum_{r=-\infty}^{\infty} \int_{-\infty}^{\infty} \text{sinc}^2[(f-\nu)N]\delta(\nu+r)d\nu \qquad \text{(B25)}$$

or

$$G_2(f) = N^2 \sum_{r=-\infty}^{\infty} \text{sinc}^2(Nf+rN) \qquad \text{(B26)}$$

Equating B21 with B27,

$$N^2 \sum_{r=-\infty}^{\infty} \text{sinc}^2(Nf + rN) = N^2 \text{ sind}_N^2(Nf) \qquad \text{(B27)}$$

or, letting Nf = u,

$$\boxed{\sum_{r=-\infty}^{\infty} \text{sinc}^2(u + rN) = S_2(u,N) = \text{sind}_N(u)} \qquad \text{(B28)}$$

## REFERENCES

1. B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill, 1969, Chapter 7.

2. A. V. Balakrishnan, "On the Problem of Time Jittering in Sampling," IEEE Trans. Inform. Theory, Vol. IT-18, pp. 226-236, April 1962.

3. H. S. Shapiro and R. A. Silverman, "Alias-Free Sampling of Random Noise," J. Soc. Indust. Appl. Math, Vol. 8, 1960, pp. 225-248.

4. A. Papoulis, The Fourier Integral and Its Applications, McGraw-Hill, 1962.